A TRAFFIC PATTERN-BASED COMPARISON OF BULK IMAGE REQUEST RESPONSE TIMES FOR A VIRTUAL DISTRIBUTED LABORATATORY

THESIS

David B. Hooten, 1Lt, USAF

AFIT/GCS/ENG/01M-03

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GCS/ENG/01M-03

A TRAFFIC PATTERN-BASED COMPARISON OF BULK IMAGE REQUEST RESPONSE TIMES FOR A VIRTUAL DISTRIBUTED LABORATATORY

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science

David B. Hooten, B.S.

1Lt, USAF

March 2001

AFIT/GCS/ENG/01M-03

A TRAFFIC PATTERN-BASED COMPARISON OF BULK IMAGE REQUEST RESPONSE TIMES FOR A VIRTUAL DISTRIBUTED LABORATATORY

David B. Hooten, B.S.
1Lt, USAF

Approved:

Michael L. Talbert, Ph.D., Major, USAF Thesis Committee Chairman	28 Feb 2001 Date
Rusty O. Baldwin, Ph.D., Major, USAF Thesis Committee Member	28 Feb 2001 Date
Karl S. Mathias, Ph.D., Major, USAF Thesis Committee Member	28 Feb 2001 Date

Acknowledgements

I would like to express my sincere gratitude to a number of individuals without whose assistance this research effort would not have been possible.

First I would like to thank Major Michael Talbert for his enthusiasm, encouragement, and guidance over the course of the past eighteen months. I would also like to thank Major Rusty Baldwin for his valuable insight and knowledge, but most importantly, the patience he displayed during my countless number of visits to his office to ask questions. I would also like to recognize Captain Paul Harmer and Eric Baenen of the sponsoring organization, the Air Force Research Laboratory (AFRL) for providing valuable input for this research effort and making themselves available on-demand.

Table of Contents

Page

Page

List of Figures

Figure | Page

Figure Page

Figure Page

List of Tables

Table Page

Table Page

AFIT/GCS/ENG/01M-03

Abstract

Various agencies throughout the Department of Defense possess intelligence imagery and electrooptical signature data required by researchers in the field of automatic target recognition (ATR). The Air Force Research Laboratory, Sensors Directorate, has been tasked with creating a virtual distributed laboratory (VDL) which will make this data available to ATR researchers via high speed networks such as the defense research and engineering network (DREN). For this research, a model for simulating potential operational network configurations and collaboration scenarios was developed and implemented using OPNET. The results of the simulations were analyzed using statistical methods to determine the impact on performance of network configuration, connection speed, server capability, and data size. Connection speed proved to be the ultimate limiting factor on system performance, but statistical insights regarding usage patterns and file sizes are drawn from the results as well. This research provides VDL designers with performance trend data and enhances the design process by providing insight into how design decisions will affect future network performance.

A TRAFFIC PATTERN-BASED COMPARISON OF BULK IMAGE REQUEST RESPONSE TIMES FOR A VIRTUAL DISTRIBUTED LABORATATORY

1. Introduction

The Department of Defense (DoD) possesses a great deal of intelligence imagery and electrooptical target signature data residing in large databases located at geographically separated government facilities across the nation. This data is used by researchers in the field of automatic target recognition (ATR) to test and evaluate algorithms designed for use in ATR systems. The Sensors Directorate (SN) of the Air Force Research Laboratory (AFRL), located at Wright-Patterson Air Force Base in Dayton, Ohio, is tasked with making these terabytes of data available to end-users at diverse locations. AFRL/SNAS has organized a Virtual Distributed Laboratory (VDL) consisting of five main parts, the algorithm developers, algorithm evaluators, collection of resources, simulation environments, and the defense research and engineering network (DREN) that ties them all together [VDL00]. Utilizing these five parts, the VDL will be able to provide anywhere, anytime, distributed database access. Furthermore, a web-based interface utilizing browsers and Java™ applets and servlets will be used to search for ATR images and retrieve those that meet the user's requirements. [WAR00].

1.1 Problem Statement

The Virtual Distributed Laboratory (VDL) is a virtual toolbox for testing and evaluating image processing algorithms using imagery and signature data held by numerous DoD agencies. In addition to the sheer volume of data holdings, many

agencies have developed metadata databases for their repositories, which describe the types of data they possess. In order to take advantage of these metadata databases, AFRL/SNAS has been tasked with implementing the vision depicted in Figure 1. The

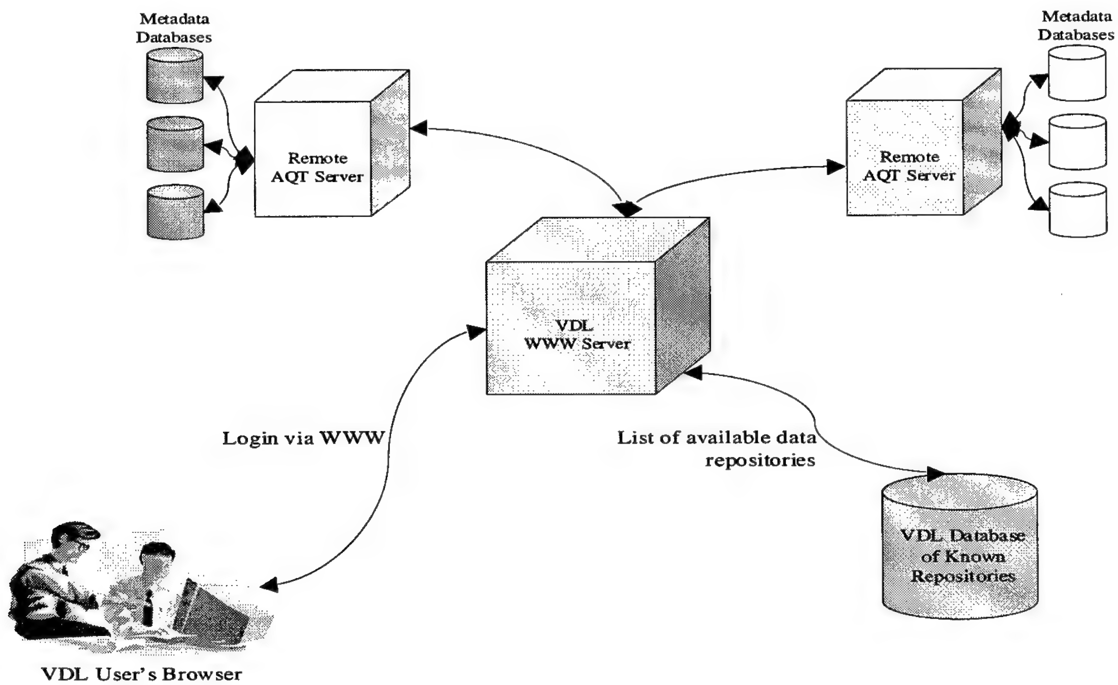


Figure 1. Vision for future VDL access

figure illustrates end-users accessing a central server and querying a database of known data repositories via the WWW. The results of the query will tell the user if the desired imagery is available and if so, the location of the data repository(s) containing the desired imagery. While research has been conducted to improve the usability of the web-based interface (Advanced Query Tool) and implement user profiling techniques [WAR00], there has been little research conducted to determine the most efficient means of getting requested data to the users of the VDL. For instance, utilizing the VDL, ATR researchers throughout the DoD will have the ability to search for and download ATR image files from remote data repositories, share information, and combine their expertise (possibly

using voice and video over the network) to develop new and better ATR systems. Additionally, the ability to utilize any one of the DoD's major shared resource centers (MSRC) for testing and evaluating complex ATR algorithms is a desired capability. Given these requirements, it is clear large volumes of data will have to pass over the VDL network. As an example, at any given time, a single researcher may request to download hundreds of megabytes or even gigabytes of data. Additionally, there may be other researchers trying to access similar quantities of data. With the potential for more than two hundred participants in the VDL, network performance quickly becomes an issue of extreme importance. Therefore, it is important to conduct research to determine what factors will have the greatest impact on the performance of the network and what improvements in the network architecture or data transfer scenario will provide the best performance.

One issue that needs to be evaluated is how the network will perform if all requests for data routed are through a central server located at AFRL/SNAS at Wright-Patterson AFB (Figure 1 illustrates this situation). Depending upon the amount of requested data and the frequency of requests; this server may potentially become a bottleneck thus limiting the usefulness of the network as a real-time collaboration enabler. This potential situation begs a question: should requested images be sent directly to the requestor for processing (potentially using up a great deal of bandwidth and creating a bottleneck at the central server) or should the processing take place on the remote server and only results sent back? A better solution might be to have the central server pass back the location of the requested data and let the requestor communicate directly with the remote server, eliminating the central server as a potential bottleneck.

Yet another scenario focuses on the ability of the network to adequately handle the anticipated amount of data traffic.

Many image files are quite large, therefore, depending upon the number of files requested and the frequency of requests; network congestion may be unavoidable. One possible solution might involve having the user send the algorithm to be processed to the server hosting the required image files. The required processing would then occur at the host server and only the results would be returned saving bandwidth and drastically reducing the possibility of congestion. This solution assumes results are significantly smaller than image files and therefore will take up less bandwidth and will reduce processing time at the central server.

Clearly, the questions posed above highlight the need to examine the best way for these systems to collaborate with one another since there are so many variables involved. Ideally, this examination will yield some answers as to the best way to configure the VDL for optimal performance thus enhancing collaboration among the participating researchers.

1.2 Goals

The primary goal of this research effort is to develop likely collaboration scenarios that accurately reflect potential VDL configurations, simulate them using a state of the art network modeling and analysis tool suite, then recommend which scenarios are most efficient for projected VDL usage patterns. Additionally, key implementation issues are examined to determine their impact on the application response time and throughput of the system. Of specific interest is the bandwidth of the connections between the user's workstation, central server, and the DREN. Statistical

analysis of the performance data obtained from varying the bandwidth of these connections will provide VDL designers with insight into the impact these varying bandwidths have on application response time and throughput.

1.3 Scope

Since the VDL has yet to be fully implemented and little measured data exists; most parameter values used in the simulations are based on predicted and planned hardware/performance characteristics. The simulations are intended to provide VDL designers with reasonably realistic performance data with which to base future implementation decisions upon.

1.4 Approach

This research effort was conducted in several phases. The first phase consisted of gathering information regarding the VDL and examining previous research. The second phase consisted of a literature review. Particular areas of focus were the VDL, DREN, the difference between distributed and parallel systems, collaborative processing, and CORBA. Knowledge obtained through the literature review was then applied in developing realistic collaboration scenarios for simulation purposes. The third phase consisted of running the simulations and the fourth phase consisted of analyzing the results. The final and fifth phase of this research effort was interpreting and presenting results with recommendations and conclusions.

1.5 Document Organization

The remainder of the document is organized as follows. Chapter 2 introduces knowledge areas required for understanding the VDL concept and developing potential

collaboration scenarios for performance modeling. Chapter 3 explains the methodology used to create and evaluate distinct collaboration scenarios and identifies the metrics used for determining the optimal scenario. Chapter 4 discusses implementation details and the results of the simulations. Finally, chapter 5 summarizes the results and makes recommendations.

2. Background

2.1 Introduction

To fully understand the methodology applied in this research effort (chapter 3), an understanding of the issues and technologies involved in the design of the virtual distributed laboratory (VDL) is needed. Furthermore, an appreciation for the role these technologies play and how they impact overall performance is important for developing reasonably realistic collaboration scenarios for modeling purposes. For these reasons, this chapter provides an overview of the main issues impacting design decisions and ultimately the performance of the VDL. Section 2.2 elaborates on the differences between distributed and parallel systems and introduces the concept of collaborative processing. Section 2.3 provides a more in-depth look at the VDL. Section 2.4 discusses the DREN network's technologies and capabilities. Finally, section 2.5 is an examination of the common object request broker architecture (CORBA). Since designers of the VDL wish to use the CORBA interface in their query tool, basic CORBA knowledge is useful [VDL00].

Understanding these areas and the roles they will play in the VDL is important to the successful development and implementation of the experiments discussed in the next chapter. For example, choosing parameters and factors that will accurately reflect possible VDL implementations is a function of how well the parameters and factors selected correlate with the actual technology/functionality being used or considered for use in the VDL.

2.2 Collaborative Processing

Parallel Processing. Prior to any discussion on collaborative processing, it is important to have a basic understanding of the differences between parallel and distributed computing. The concept of parallel computing is easy to explain. Borrowing from an example Kumar uses in his book [KUM94], a library is used to illustrate the concept. The task is to shelve all the books in a library in the proper order. With only one worker to accomplish this task, it is going to take a fixed amount of time. Now consider multiple workers, say one per bookshelf, performing the same task. All the workers are now shelving books simultaneously. When a worker finds a book belonging to another shelf, that book is passed on to the worker at that shelf. While this example is over-simplified for the sake of illustrating the concept, it should be intuitive that the task will get done much faster with multiple workers as opposed to just one worker. The same concept can be applied to computer processors. In many cases (depending upon the task), several processors working together simultaneously to solve a large problem can do it faster than one processor working sequentially. As defined by Foster, a parallel computer is a set of processors that are able to work cooperatively to solve a computational problem [FOS95]. Although situations do exist where parallel processing is not the best solution (e.g., small computations where the communications overhead far exceeds the processing time), the concept is important to this research effort. Many of the automatic target recognition (ATR) algorithms designed by researchers who will ultimately use the VDL, require parallel processing systems to run. This means many algorithms will have to be run at one of the DoD's high performance computing centers

(HPCs). This fact has a significant impact on design decisions and therefore must be known to anyone doing VDL-related research.

Distributed Computing. Tanenbaum defines distributed computing as, "a collection of independent computers that appear to the users of the system as a single computer." [TAN95] This is the definition used for the remainder of this research effort. There are two major aspects of distributed systems:

1. The computers in a distributed system are autonomous (hardware).
2. The user thinks of the system as a single computer (software).

First, unlike parallel systems, which operate in a homogenous environment, distributed systems operate in a heterogeneous environment. For example, a Windows NT machine may communicate with a UNIX-based system for purposes of file sharing. Machines in a distributed system can communicate regardless of hardware or operating systems employed. The second aspect deals with the concept of transparency. On a network where files are stored on a network file server (NFS), when a user accesses these files, they appear to be on the user's local drive. Another example is a network printer. When a user elects to print out a document, the user does not have to know the printer is located in another room or attached to another computer. All that matters or is visible to the user is whether or not the document printed or not. This is what is meant be transparency. Everything appears as one system to the user when in fact the resources being used are distributed. [TAN95]

Collaborative Computing. With the distinction between parallel and distributed computing made, the concept of collaborative computing can be examined. To collaborate is defined by the American College Dictionary as, "to work, one with

another; cooperate, as in literary work." Applying this definition to the field of computers, collaboration must mean computers working one with another, cooperating. While this definition seems intuitive, for purposes of this research effort, a more concise definition is required. The most concise definition for collaborative systems found comes from Farley. He states, "*A collaborative system is one where multiple users or agents engage in a shared activity, usually from remote locations. In the larger family of distributed applications, collaborative systems are distinguished by the fact that the agents in the system are working together towards a common goal and have a critical need to interact closely with each other: sharing information, exchanging requests with each other, and checking in with each other on their status.*" [FAR98] A term used to describe systems that utilize collaborative processing is "collaboratories." This term stems from the realization that by combining the interests of the computer science and engineering community with those of the scientific community, laboratory and technical research can be carried out effectively without regard to geographical separation. [SUP00]

Collaboratories have become extremely important for several reasons, the two most important being discussed next. First and foremost, the major impediment faced by researchers today is geographical separation. The separation can become an impediment to effective information sharing and cooperation due to cost of travel and time differences. Second, it is not uncommon for sophisticated problems to be worked on by teams of scientists pooled from various universities, national laboratories, and industry. These researchers need the ability to communicate their findings with one another, share data and even instrumentation regardless of the geographical separation or the types of

networks or computers being used in the research. In their article, "Distributed, Collaboratory Experiment Environments (DCEE) Program: Overview and Final Report," Johnston and Sachs describe the vision for distributed collaboratories as follows: *"..to provide a widely distributed environment in which people, instrumentation, and information can flow and interact as easily as they can when all of the critical resources are local.*" [GEO00]

Combining all of these concepts, parallel processing, distributed processing, and collaborative processing, the interrelation of concepts behind the vision for the VDL is complete. The VDL will be a collaboratory. Researchers from throughout the DoD will be able to able to query a central server from a remote location and find out where specific types of data can be found, run algorithms against this data, and share results. As a whole, the system will be distributed and the process of finding data and running algorithms will be transparent to the user. Parallel processing will be a function of the HPCs. When large complex problems need to be run, parallel systems at one of the HPCs can be utilized.

2.3 VDL Central Library

The VDL consists of five main parts, the algorithm developers, the algorithm evaluators, a collection of resources, simulation environments, and the DoD's high speed networks. Figure 2 illustrates all of these pieces interacting with each other to accomplish the mission. The resources piece consists of several sub-pieces including the VDL Central Library. The other pieces are the DoD data repositories and HPCs, also

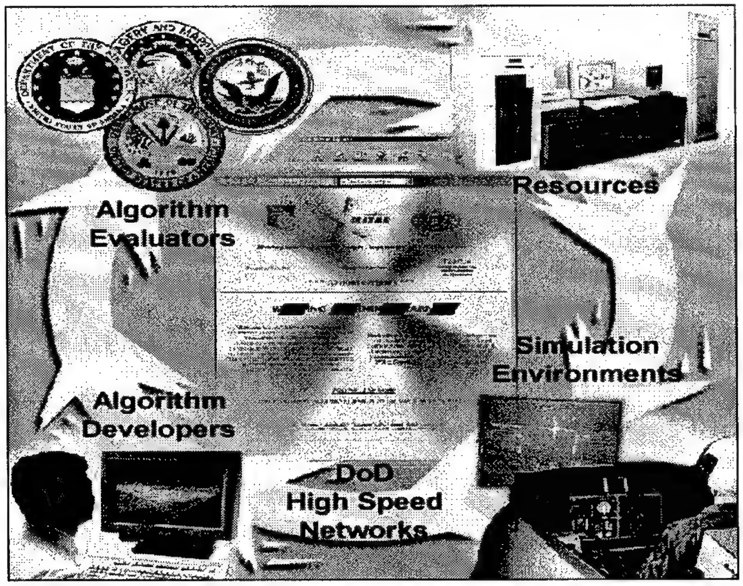


Figure 2. VDL – the big picture

known as major-shared resource centers (MSRC). An excerpt from the AFRL web site summarizes the purpose of the VDL Central Library: "*...the VDL Central Library is a toolbox designed to support algorithm evaluators, imagery/signature data collectors and users, and researchers and developers across all of the Department of Defense (DoD) in the fields of ATR, information/sensor fusion and C4ISR. The VDL Central Library will continually evolve to provide services and resources for the DoD community.*" The following four sections contain descriptions of the remote image query tool, the information library, algorithm evaluation, and information sharing. [VDL00]

2.3.1 Remote Imagery Query Tool

In order to effectively develop, test, and evaluate image-processing algorithms, imagery or signature data is required. Many agencies throughout the DoD working with automatic target recognition (ATR), Fusion, and C4SI have collections of this data that are sometimes stored off-line on tapes or disc or on-line in databases. Additionally, some of these agencies have created meta-data databases which are databases containing

records which describe the types of data in the collection. To date, a major problem plaguing researchers throughout the DoD in the field of ATR research has been determining what data specific agencies possess and if that data is of any use to a given project. This problem is solved with the remote imagery query tool (RQT). The RQT will take as input a user query or description of the type of data required and will return the location of the data regardless of where the data physically resides within the DoD. The following paragraph describes desired functionality of the RQT. [VDL00]

The RQT will utilize a web interface and will contain a form that the user will fill out to indicate the parameters of the data required. Figure 3 provides a snapshot of the

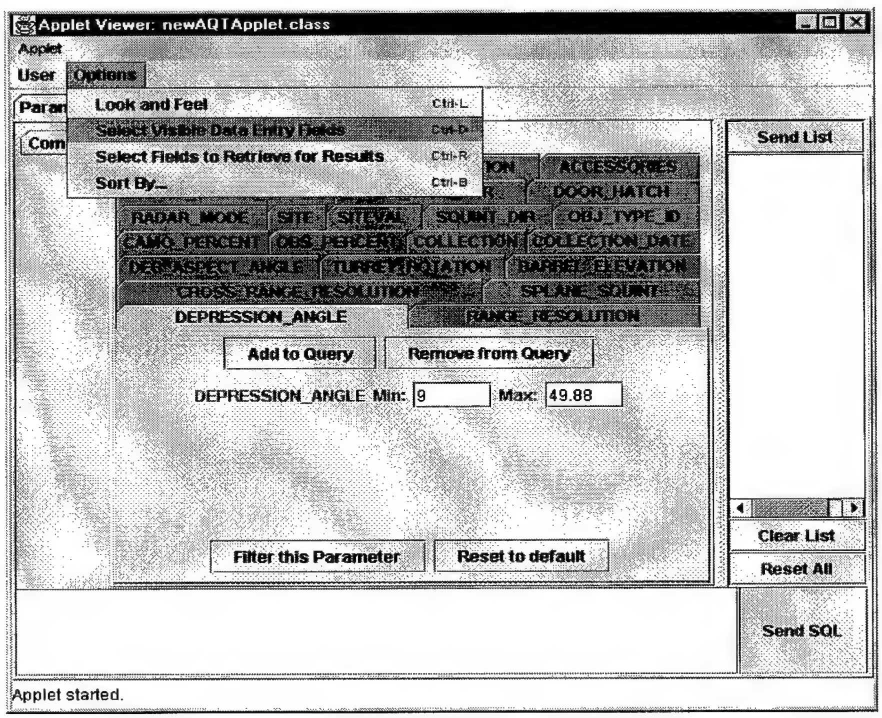


Figure 3. AQT 2.0 user query interface

Advanced Query Tool 2.0 user query interface, which is the most recent iteration of the RQT. Once the form is completed, a query will be sent to a central server which will then query a database of known data repositories (both the central server and the database

of known repositories are part of the central library). Upon receiving the results of the query, the central server will send the results back to the user. At this point, the user will have to make a choice. Depending upon the number of images available and whether or not the data is stored on-line, the user can either download all of the images or a random sampling of them (for example, the user may only wish to download 200 out of 3,000 available). If the requested data is only available off-line on media such a tape, CD, or DVD, then the user can fill out a data request form to have the data shipped. [VDL00]

An important software component of the RQT being developed is the data/imagery phone book. This phone book will be a database containing generalized information about the various data repositories throughout the DoD as well as planned future data repositories. The phonebook can be used independently of the RQT and will provide information regarding both on-line and off-line data as well as data that may be at a security classification different from the user's network. Additionally, the phonebook will provide links to other on-line sources of the requested data as well as data descriptions and points of contact. [VDLCL]

The desired functionality of the RQT presents designers of the VDL with some challenges regarding performance. Currently, the most pressing issue is being able to provide the user with the ability to download data sets (image files) from any location. Some of these data sets may contain hundreds or even thousands of images (megabytes to gigabytes worth of data). Depending upon available bandwidth, connection speeds, number of desired files, amount of network traffic, etc., this may lead to a significant amount of network congestion since image files can be quite large. Additionally, given the intended design, it would appear the potential exists for a bottleneck at the central

server since the primary traffic over the network will consist of image files. Other factors potentially impacting performance of this design are the number of users requesting data at any given time and the frequency of requests. Clearly, understanding the desired functionality of the RQT is important in designing realistic collaboration scenarios for simulating.

2.3.2 Information Library

Due to the geographical separation of those performing research in the field of ATR, information fusion, and C4ISR (Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance), collaboration is difficult at best and leads to duplicative effort. For example, suppose two separate organizations have developed similar algorithms (duplication of effort). Although one algorithm may be considerably better than the other, since the two organizations are not aware of each other's efforts, they cannot compare their algorithms or share information. If these organizations were to combine their efforts, or at least share information, they may be able to develop algorithms that perform better than those already in existence. This is one of the reasons why collaboration is so important and also serves to highlight the need for a centralized information library. [VDL00] The next two sections detail the main objectives associated with the information library.

There are two main objectives. One is to allow users to select two or more image processing algorithms, have a third party, "an honest broker", evaluate them and return a set of standardized results. Objective two is to expedite the sharing of ATR, information fusion, and C4ISR-related data on a DoD-wide basis. What is desired is essentially a one-stop-shop for any ATR, information fusion, or C4ISR data requirement [VDL00].

2.3.2.1 Algorithm evaluation

In order to meet this objective, the third party or “honest broker” would be required to perform the following duties:

- download algorithms and install operating instructions,
- download a standardized evaluation plan,
- download a standardized evaluation data set,
- download a standardized evaluation results report template,
- download any relevant evaluation metrics documents,
- perform evaluation, and
- post results [VDL00].

Given these requirements, it appears the central server will potentially experience a substantial amount of data requests and dissemination traffic, especially when taking into account a large amount of these downloads will consist of image files. Again, the question arises, is this the best scenario? One alternative scenario involves allowing the site where the majority of the evaluation data (image files) resides perform the evaluation assuming they have the computing resources. In this manner, far fewer image files would have to be sent over the network greatly reducing the potential for network congestion and server-related slow-downs. The centralized processing scenario, discussed in the next chapter, is designed with this concept in mind.

In addition to downloading image files, researchers using the VDL will be able to share information with one another. The next section lists the types of information that will be available to ATR researchers as a result of the information sharing capability the VDL will provide.

2.3.2.2 Information sharing

The types of data stored in the information library will include but are not limited to the following:

- algorithms (source code),
- documentation (including installation instructions),
- design documentation,
- standardized test and evaluation plans,
- standardized test and evaluation data sets,
- standardized test and evaluation methodology,
- test and evaluation metrics documentation,
- standardized test and evaluation results reporting templates,
- evaluation results,
- technical and white papers, and
- any other information that may be useful to the DoD community [VDL00].

2.4 Defense Research and Engineering Network (DREN)

The DREN is a high-speed network which links approximately 60 DoD research and development facilities throughout the lower 48 states, Alaska, and Hawaii [DYK00]. All of these facilities will have access via the DREN to the DoD's HPCs for purposes of fulfilling computational requirements and expediting algorithm development [DREN 00]. In order to be effective, the DREN must deliver performance similar to that of the HPCs with which they will connect. To meet these performance requirements, the DREN will provide Internet Protocol (IP) and Asynchronous Transfer Mode (ATM) services ranging from 10Mbps through Gigabit/sec speeds [DYK00].

2.4.1 IP addressing

A critical part of any communications network is the protocol with which one machine communicates with another. With IP, the address format is specific. IP uses the host/network address scheme. A given computer on an IP network possesses a host name and a network or IP address. Utilizing either the host name or the IP address, messages can be sent to a particular machine on the network. As an example, the JavaSoft home page exists on the host named www.javasoft.com and has an IP address of 204.160.241.98 [FAR98].

2.4.2 Asynchronous Transfer Mode

Over the past few years, asynchronous transfer mode (ATM) has gained popularity for four major reasons; interoperability, standardized transmission protocol, one network for all information requirements, and various speeds for various users [ATM00]. Each one of these areas will be examined in detail in the next few sections, but first a list of ATM characteristics is provided as a primer for the discussions that follow.

2.4.2.1 Asynchronous Transfer Mode characteristics

Listed below are the primary characteristics and advantages of ATM. These characteristics and advantages are required knowledge for fully understanding the discussions in the next four sections of this chapter. Additionally, these characteristics are important in following chapters where the design and simulation of ATM network models are discussed.

Characteristics of ATM

- Efficiently transfers video, audio and data,
- Bandwidth can be allocated as needed (1.54Mbps – 622.08Mbps),
  - T1 & DS1 = 1.54 Mbps
  - T3 & DS3 = 44.7 Mbps
  - OC3 = 155.5 Mbps
  - OC12 = 622.08 Mbps
- Fixed-length packets of 53 bytes are used, 5 bytes for the header and 48 bytes for data. Additionally, the packets are guaranteed to arrive in order.
- ATM is connection-oriented, that is it uses a virtual circuit to transmit packets that share the same source and destination over the same route [IUK00].

Advantages of ATM

ATM networks are ideal for the VDL since they offer the following advantages:

- *Support business process re-engineering* – the exploration of new telecommunications capabilities. Allows an organization to stay ahead of competitors.
- *Improve the flow of information* – accurately and timely delivers data.
- *Fast communications for decentralized organizations* – remote employees accessing the same resources and tools.
- *Provide communication linkage for effective collaboration* – many people from around the globe can come together electronically on a case-by-case basis to solve problems or develop new products.

- *Can speed up market response and product development* – the ATM infrastructure allows organizations to respond quickly to changing conditions, collaborate on new projects, and implement changes [GAD97].

2.4.2.2 Interoperability

Interoperability is an aspect of the emerging requirement for distributed and collaborative processing. As more and more information is becoming available in on-line digital libraries, information must be available regardless of the type of system used or information being requested. Heterogeneous systems must be able to share data.

2.4.2.3 Standardized transmission protocol

One of the major problems plaguing the network industry has been that of various transmission methods/protocols. Typically, the transmission protocol used for a LAN is different than that for a WAN. This poses problems as user needs expand. As opposed to only communicating with systems within a given network (LAN or WAN), computers now need to communicate on a world-wide scale. ATM is a good solution for this problem because it is well-suited for both LAN and WAN technologies. [ATM00]

2.4.2.4 One network

In many cases today, separate networks are used to transfer different types of information such as data, voice, and video. This is done because these different data types have different characteristics. For example, data traffic is bursty whereas voice and video traffic need to communicate for extended periods of time and are more evenly distributed. Another important aspect of voice and video is the importance of the order the information arrives. If the information arrives in a different order than it is shipped, then the voice or video information will be distorted or totally useless. This is not a

problem with ATM since ATM packets are guaranteed to arrive in order. As a result, if utilizing ATM, there will be no need for separate networks for the different types of data. ATM was designed from the beginning with this in mind and can accommodate simultaneous transmission of all three types of data. [ATM00]

2.4.2.5 Tailored performance

The final advantage of ATM is the dynamic allocation of speeds ranging from

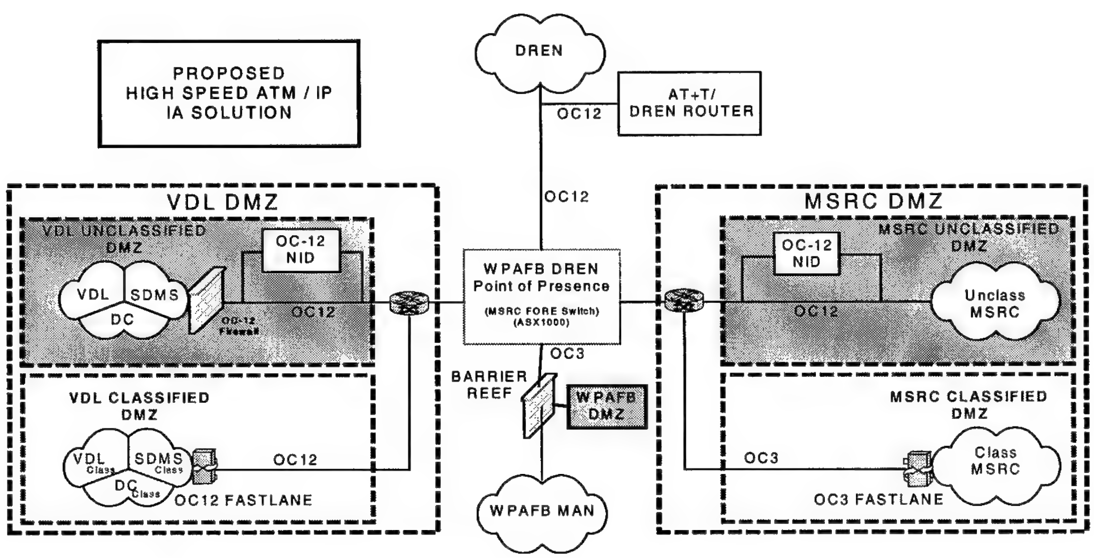


Figure 4. Proposed ATM/IP high speed solution for VDL [VDL00]

1.54 Megabits/second (T1) to 622.08 Megabits/second (OC12). This preserves bandwidth for the users who require it without degrading performance for themselves or users with lower bandwidth requirements. [ATM00] With this basic knowledge of ATM, an examination of the proposed ATM/IP high speed solution (Figure 4) for the VDL can now result in the creation of more accurate models for simulation purposes.

2.5 Common Object Request Broker Architecture (CORBA)

CORBA is a specification developed by members of the Object Management Group (OMG), a consortium of over 700 companies, for building and using distributed

objects. The CORBA specification is based on the abstract object model defined by the OMG. The model is abstract because while it does specify a standard way for using objects, it is technology independent. In other words, objects can run on any platform, be located anywhere on a network, and can be implemented in any programming language provided they adhere to the CORBA specification [MAH00]. The CORBA architecture consists of five major components, the object request broker (ORB), interface definition language (IDL), dynamic invocation interface (DII), interface repositories (IR), and object adapters (OA) [MAH00]. These components will be discussed below. Following the discussion of the five components of the CORBA architecture, distinctions between CORBA and Java RMI will be examined.

2.5.1 Object request broker (ORB)

The ORB is the software that implements the CORBA specification and is the center of the CORBA model. The ORB allows a client to communicate with a server when dealing with distributed objects. Both the client and server must communicate with each other via the ORB. [FAR98]

The ORB is responsible for the following tasks:

- Finding the object implementation for the request,
- Preparing the object for receiving the request, and
- Communicating the request.

Regardless of whether the client and server are on the same machine or are separated by a network, all requests must be handled by the ORB [MAHMOUD00]. When the ORB receives a request from the client, it searches for the implemented object in the distributed system. When found, the ORB will use the client's skeleton interface to

invoke the implemented object and will generate a language-specific form or stub the client can then use to invoke a method on the remote object [FAR98]. The CORBA ORB architecture is depicted in figure 5. Unless the client and server are implemented on the same machine, each will have the same components of the CORBA ORB architecture shown in the figure.

2.5.2 Interface Definition Language (IDL)

The "implemented object" has an interface that defines what operations the object can perform and the parameters to those objects. The interface is defined by the IDL and is the contract between the client and server [MAH00].

With an interface defined, any programming language that has IDL mapping can be used to make requests to the object provided the requests adhere to the interface. Likewise, with a defined interface, a given object can be implemented in any appropriate language. Some languages that have IDL mapping are C, C++, Java, Smalltalk, and Lisp [MAH00].

2.5.3 Dynamic invocation interface (DII)

Stubs are the way in which clients could invoke methods on remote objects. Client stubs are created using static interfaces -- interfaces that are determined at compile time. Another option is to use dynamic interfaces. Dynamic interfaces allow client applications to use server objects without having any knowledge of those objects at compile time. The client can simply obtain an instance of the object and then dynamically make requests on that object. The DII simply uses the interface repository (discussed in the next section) to validate the client's request. CORBA supports both static and dynamic interfaces [MAH00].

2.5.4 Interface repository (IR)

Without any compile-time knowledge of object interfaces, the client has to have a way of determining how to interface with available objects. This is the purpose of the IR.

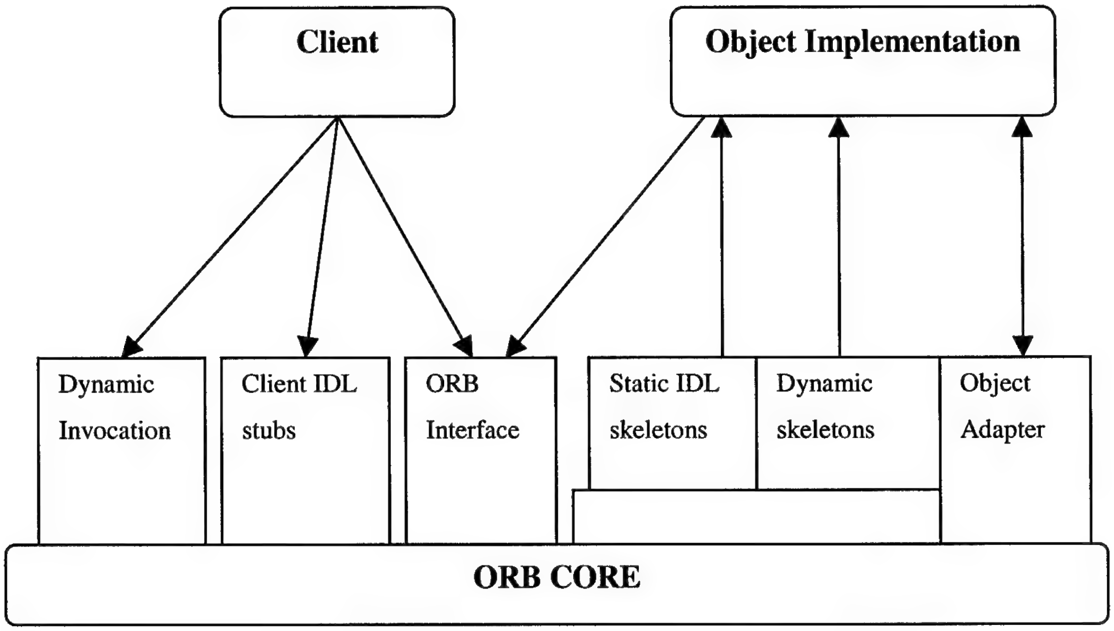


Figure 5. CORBA ORB architecture [MAH00].

The IR contains interfaces to various objects that the client can use to construct requests. Once the request is built, it can then be forwarded to the ORB. The IR facilitates DII [MAH00].

2.5.5 Object adapters (OA)

Object adapters are the way in which an object implementation accesses the services of the ORB (see figure 5). Mahmoud lists the following ORB services as those accessed via the object adapter:

- Object reference generation and interpretation.
- Method invocation.
- Security of interactions.

- Object implementation and activation and deactivation [MAH00].

Other distributed object systems are available. One of those systems is Java RMI or Java remote method invocation. The following section provides a brief comparison of CORBA and Java RMI.

2.5.6 CORBA vs. RMI

Like CORBA, Java RMI is a distributed object system. The main difference lie in the fact that for any two systems to communicate using RMI, both must have their applications programmed in Java. In other words, Java RMI is language-dependent. [FAR98] Farley and Mahmoud both list differences between the two implementations. Below is a composite list:

- RMI is easier to master. CORBA is more complex and it may be overkill to learn the specification depending upon the task at hand.
- CORBA is language-independent and can run in heterogeneous environments whereas RMI requires a homogeneous language environment to operate in (Java).
- CORBA is a mature standard and is more robust.
- RMI is cross-platform. Any distributed object in RMI can be relocated on any other host in the system. CORBA does not support this. CORBA implementations must remain on the host they were created on. They can only send references to themselves to other objects [MAH00] [FAR98].

Clearly, both implementations have their advantages and disadvantages. Deciding on one or the other depends on the environment in which the implementation will be running. For example, if a system is being built from scratch and there are no legacy systems

involved, Java RMI may be the best alternative so code portability and Java features such as serialization can be capitalized on. On the other hand, if the system were to include legacy systems with peculiar needs, CORBA would be the best solution since it is language independent. There are some languages such as C that are better suited than Java for handling computationally complex problems. For this reason, the system may need to maintain its language independence. Knowledge of the advantages and disadvantages of both distributed object systems is important. Possessing this knowledge gives designers of the VDL more latitude when making design decisions. Furthermore, if they can project future requirements, determining which implementation is best for the long run is made easier.

2.6 Summary

This chapter provides basic knowledge required for understanding the need for the VDL and the approach researchers at AFRL/SN are taking to fulfill that need. Additionally, an understanding of the VDL, from its inception to its current state, is important when designing experiments to simulate the performance of the VDL network. These experiments, their design, implementation, and results, are discussed in the next two chapters. In summary, this chapter first discussed the distinction between parallel and distributed processing as well as the concept of collaborative processing. Next, a more extensive look into the VDL concept was provided. Following this was a discussion on the DREN and its capabilities. Finally, CORBA was examined and compared to Java RMI, another option available to designers for implementing distributed objects.

3. Methodology

3.1 Introduction

The scenarios outlined in this chapter were simulated using a modeling tool called OPNET Modeler. The primary purpose of the simulations is to demonstrate to designers of the VDL the performance advantage of one scenario over another. To accomplish this, throughput and application response times are to be measured and compared. Additionally, within each scenario, selected factors are manipulated to determine their impact on the throughput and application response time.

The remaining sections of this chapter describe the methodology used in conducting this research. Sections 3.2 though 3.15 consist of discussions regarding the three collaboration scenarios evaluated, custom application, system boundaries, system services, performance metrics, parameters, factors, evaluation techniques, workload, experimental design, and the chapter summary.

3.2 Baseline scenario

The baseline scenario for this research effort is based upon the envisioned VDL architecture (which will henceforth be called the baseline architecture) introduced in chapter 1. Figure 6 shows the baseline scenario. In this scenario, the user submits queries to the central server requesting specific images based upon the parameters specified in the query. The central server performs a search of a database of known participating data repositories throughout the DoD to determine if the requested images exist. The results of this search (number of images, file names, etc.) are sent back to the user who decides which images to download. Once this decision has been made, the user

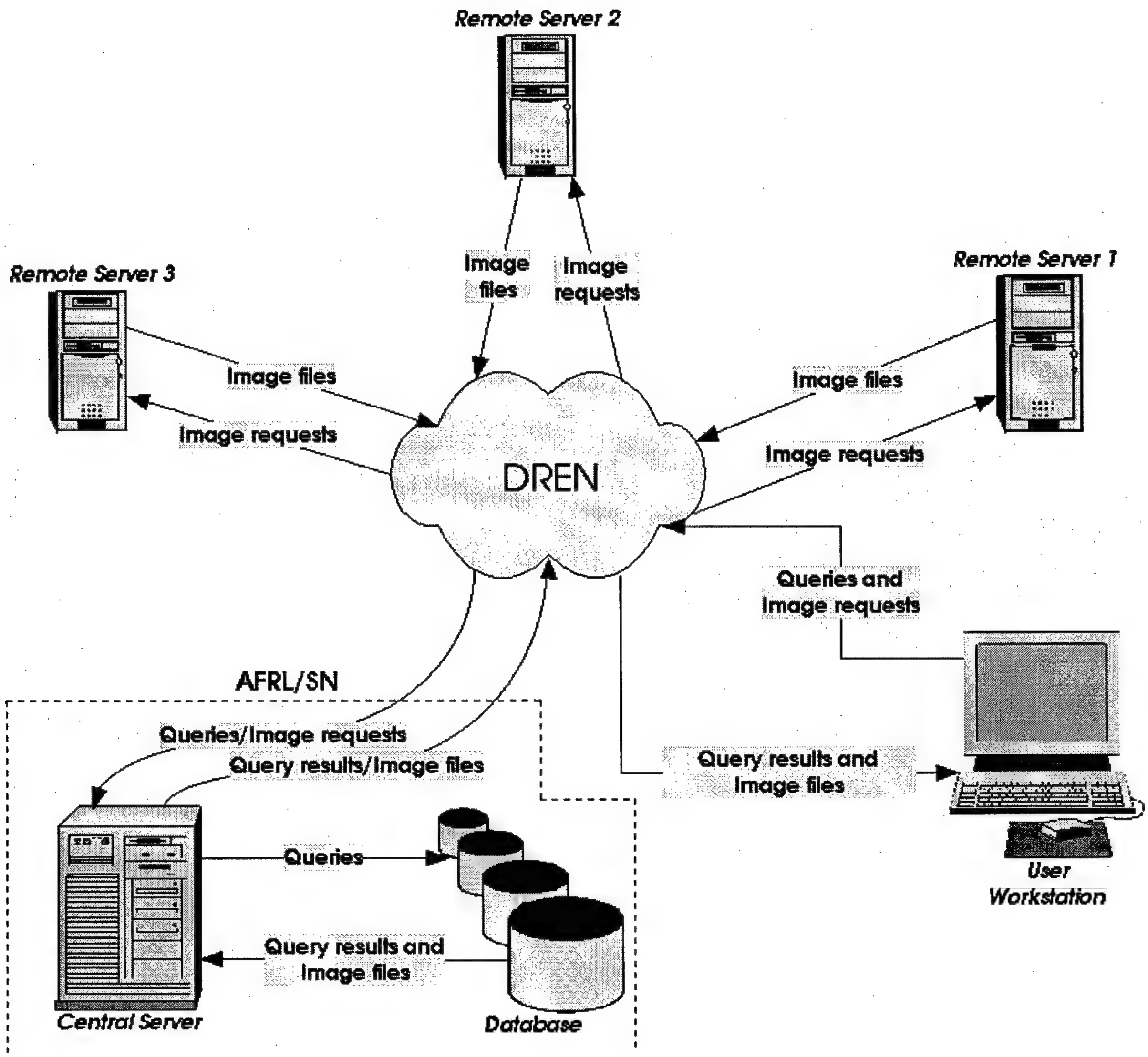


Figure 6. Baseline scenario

requests the central server to retrieve the images. The central server will either fulfill the image request or will acquire the requested images from remote sites on behalf of the user submitting the request. As the images are retrieved, they will be routed back to the user. The user can then process an automatic target recognition (ATR) algorithm on the images locally as desired. It is assumed that the user has the required computational resources for processing algorithms using the downloaded images. The main drawback to this scenario is it requires the transmission of very large image files (in the Megabyte

range or even greater) over the network, which could quickly overwhelm the central server resulting in severe congestion.

3.3 Scenario 2 (Centralized processing and image storage)

An alternative scenario emphasizing centralized algorithm processing is depicted in Figure 7. In this scenario, AFRL/SN is assumed to possess a database with copies of all ATR images known to exist rather than acquiring them from remote locations. This would eliminate the need to transmit large image files over the network (with the exception of occasional updates to the database, which would occur infrequently). In addition to the image database, it is assumed AFRL/SN possesses a major shared resource center (MSRC) which has the required computational resources for processing ATR algorithms. As in the baseline scenario, the user will still query the central server to determine what images are available and will choose those that are desired. However, instead of downloading those images, the user will send the algorithm(s) and associated documentation and tools to the central server for routing to AFRL/SNAS's MSRC for processing against the selected images (which are transferred from the database to the system performing the processing). Once the algorithm processing completes, results are sent back to the user through the central server. This scenario differs from the baseline scenario; no image files are sent over the network. Only the algorithms and results of the processing are being transmitted over the network. Although the algorithm files and result files can be quite large (algorithm packages can be as large as 3GBytes and results can be as large as 10MBytes), they only get transmitted once as opposed to a user downloading hundreds or even thousands of ATR image files [BAE00]. The hypothesis for this scenario is that two data transfers of very large files (up to 3 GB) will still provide

better throughput and application response time over the transfer of many ATR image files.

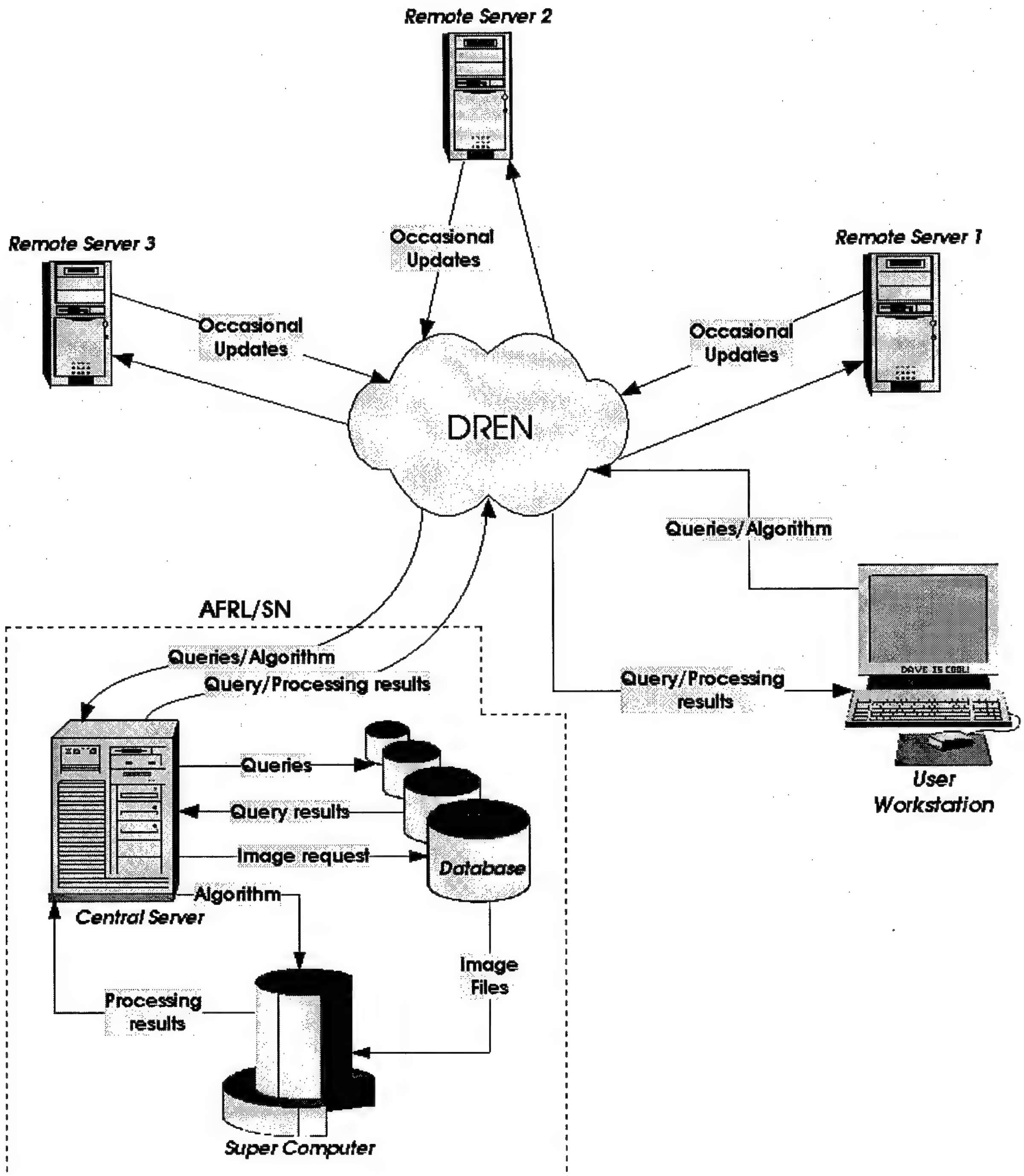


Figure 7. Scenario 2 (centralized image storage and processing)

3.4 Scenario 3 (Direct download from remote site)

Scenario 3 is similar to the previous two scenarios because users still submit queries to the central server to locate ATR images. The difference is in the way the user will acquire the image files. When a user receives from the central server the list of image files meeting the query parameters, the list will also point the user to the location of the files. Instead of the user submitting a download request to the central server, the user will download the required image files directly from the data repository where they reside (see Figure 8). Although this scenario does not eliminate the flow of image files over the network, it does eliminate the transfer of image files from remote data repositories to the user via the central server. The hypothesis being tested in this scenario is that having the user directly download the image files will provide better throughput and application response time compared to the baseline scenario.

3.5 Other considerations (Factors)

In addition to considering the impact different file sizes and traffic patterns have on the performance of the network, data rates are examined. Specifically, the data rates of the connections between the user's workstation and the DREN access point (ATM switch) and the central server and the DREN access point are of primary interest. The data rate of the user's connection is an important aspect of the network to examine since all users do not necessarily have the same connection speeds. Some users may be limited to T1 (1.54 Mbps) data rates while other users may have T3 (44.74 Mbps) data rates or higher. For this reason, the data rate of the connection between the user's workstation and the DREN access point is varied between T1 and T3 during the simulations. Another important connection data rate to examine is that of the connection between the central

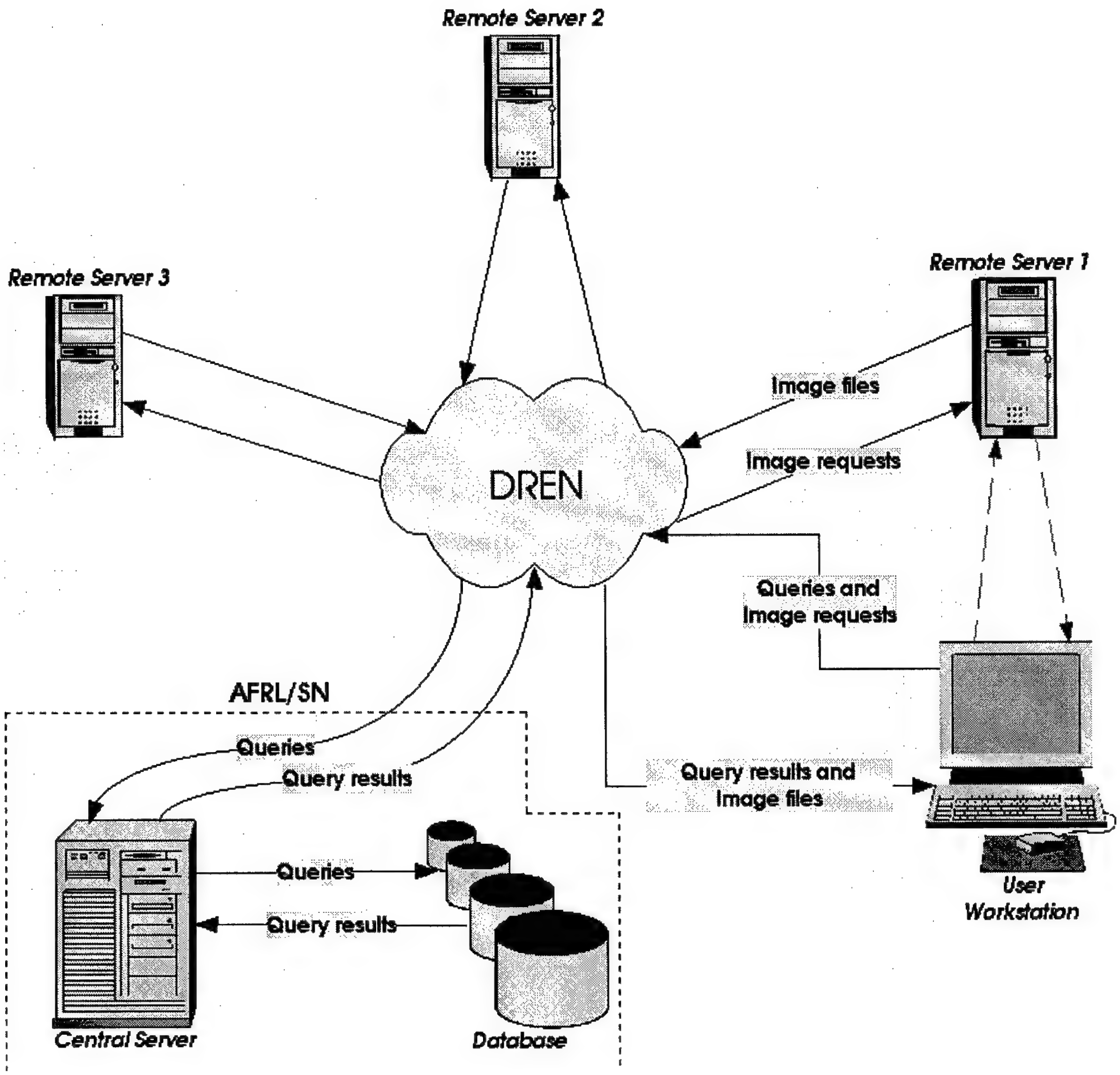


Figure 8. Scenario 3 (direct download from remote sites)

server and the DREN access point. Currently, this connection is limited to a data rate of 8 Mbps as a result of having to share bandwidth with the rest of the installation's organizations. All connections to the outside must pass through the installation's barrier reef for security purposes. To improve performance, designers of the VDL would like to have a dedicated link between the central server and the DREN access point with an

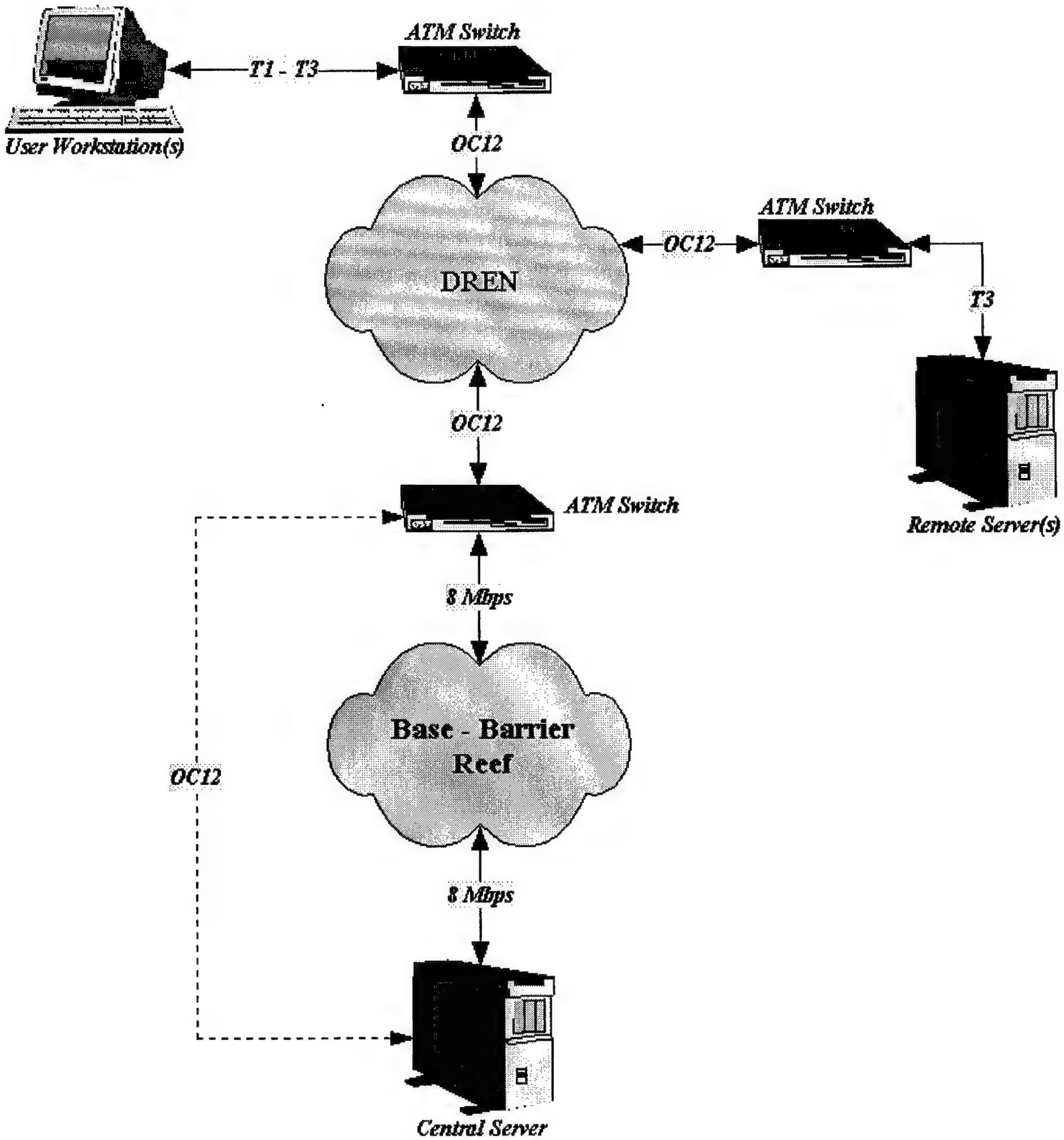


Figure 9. Connection data rates

OC12 (622.08 Mbps) data rate. This link would completely bypass the barrier reef thus theoretically providing much better performance. For this reason, the central server ←→ DREN data rate is factored into the simulations and will be varied between 8 Mbps and OC12 to determine the magnitude of improvement in performance (throughput and application response time). The data rates of the connections between the remote servers

and the DREN will be set at T3 for all simulations. Likewise, the DREN data rate (ATM switch to ATM switch) will be set at OC12 for all simulations. Figure 9 shows the various connections and their associated data rates.

3.6 System Boundaries

Simulating the VDL requires a comprehensive understanding of the components making up the VDL as well as how those components interact with each other to fulfill the user's request. This entire section is dedicated to providing the necessary information required for understanding the system being tested, the components that make up the system, and the role each component plays within the system.

3.6.1 System under test (SUT)

The system under test (SUT) for this research effort consists of the VDL network and all associated components. The components of the SUT are servers, workstations, databases, interconnecting network, and the DoD major-shared resource centers (MSRCs). All of the components listed will play a part in the implemented VDL, however, not all of them will be factors in the simulations. For example, in the second scenario where the processing takes place at an MSRC, the amount of time it takes the MSRC to actually start a job and process it is not considered since it has no bearing on the bandwidth or data rates obtainable over the network. The primary interest is how the network and servers handle the traffic being sent to and from the MSRC. For this reason, this particular simulation will be run with the central server providing the same processing capabilities as it did in scenarios 1 and 3 (specifics are provided in chapter 4). Likewise with the databases, the simulations do not factor in the time it takes a server to execute a query in a database and receive results. While these actions must occur, they

do not play a role in the simulations since no DREN or other data "pipes" are used. Follow-on research can be conducted to examine these issues in more detail if so desired. The following five sections describe in more detail each component's role in the VDL and how they are simulated in OPNET.

3.6.1.1 Servers

There are two types of servers being used in the VDL, the central server and remote file servers. The central server has several responsibilities. First, it processes all image queries submitted by the users. When a query is received, the central server will submit the query to a database then route the number of files meeting the query parameters back to the user. The central server also processes download requests. For example, in the baseline scenario, when a download request is received, the central server will retrieve the images from the image database and send them back to the user. If the images are not available in the central library, the central server will then forward the request to the appropriate remote server(s) for processing. When the central server receives the requested images from the remote server(s), they are then routed back to the user. In the centralized processing scenario the user does not download images. The central server will receive an algorithm from the user, route it to the MSRC for processing using the user-selected images, then send the results of the processing back to the user. Finally, in the direct download scenario, the user will download the requested images directly from the remote sites. The central server is left out of the picture completely unless the central server has access to some or all of the desired image files.

Remote servers simply act as a gateway to the site's data repository. When a download request is received either from the central server or directly from the user, the

remote server queries the image database for the requested files and sends them to the requestor. Each remote server will run an advanced query tool (AQT) interface, which allows any registered VDL user to access the site regardless of differences in hardware or operating systems.

3.6.1.2 Workstations

Workstations are simply the machines VDL users are utilizing to access the VDL. In the simulations, they represent the point of origin for user requests. The workstation is where the user submits requests and receives results/image files. Additionally, statistical data such as application response time is collected at the workstation node.

3.6.1.3 Databases

Databases store data ATR researchers find useful. Examples are ATR image files, location and source information, results, and miscellaneous documentation. Clearly, they are an integral part of the VDL. One typical use of the database involves the central server. The central server accesses a database of known data repositories to determine if the image files requested by the user exist and if so, where. Additionally, all servers in the VDL, including the central server, must access databases to retrieve image files tagged by the user for downloading. While databases don't actually factor into the simulations (access times are not being considered), understanding where they fit into the overall scheme is important, especially for future performance evaluations where database access times may be considered.

3.6.1.4 Interconnecting network

The interconnecting network simulated in the experiments is an ATM network with data rates ranging from T1 to OC12. The DREN portion of the interconnecting

network is modeled with an "ATM cloud" node that simulates the behavior of an entire ATM network. All links in the models are "ATM link" nodes and have adjustable data rate attributes. Since link data rates from the workstations and servers are different from the DREN data rate, ATM switches were added to allow for separate links with different data rates. This is required since link speeds are varied during the simulations.

3.6.1.5 Major Shared Resource Center (MSRC)

MSRCs provide the computational power required for solving large problems. They include systems such as workstations, networks of workstations and servers, parallel systems, and mass storage systems [HPC00]. MSRC processing times are not factored into the simulations; however, it is important to understand where they fit in since future research may factor in the processing delays associated with running jobs at an MSRC. The only scenario that involves the MSRC is scenario 2 (centralized processing). In this scenario, the user is taking advantage of the computing resources available at the MSRC. It is assumed for this research effort that any given MSRC can process any algorithm and amount of data sent to it. For the first and third scenarios, it is assumed the user has the required computing resources available locally.

3.6.2 Component under test (CUT)

The component under test is the interconnection network. Focus is on evaluating the impact traffic patterns, file sizes, and connection data rates to the DREN have on system throughput and application response time. For all experiments conducted, the DREN provides an OC12 data rate. Additionally, the same server model, workstation model, and associated parameters were used in the simulations, therefore, the only factors changed from one simulation to the next were traffic patterns, file sizes, link background

utilization, and data rates obtainable over the user and central server connections to the DREN.

3.7 System Services

The SUT simulated in this research effort provides the following services:

- Distributed access to large data repositories (search and download capabilities).
- Distributed access to powerful computing resources (MSRCs).
- High-bandwidth capability for transmission of large amounts of data.

3.8 Performance Metrics

The performance metrics of primary interest are *throughput* and *application response time*. Both of these metrics are recorded during the simulations and used to compare the performance of the three scenarios. Detailed discussions of both metrics are provided in the next two sections.

3.8.1 Throughput

Throughput is defined as the rate (requests per unit of time) at which requests can be serviced by the system [JAI91]. Throughput is a required "higher-better" metric. From a performance standpoint, the rate at which the system can service the requests is extremely important. In the scenarios previously discussed, a request is considered fulfilled each time the user receives back an image file or processing results. The total throughput for the system is then calculated by dividing the time required for downloading the requested files by the number of files requested.

There are several factors, which will affect this throughput value. The most obvious is the bandwidth of the various links in the network. In any given circuit, from source to destination, the effective throughput will be limited to the link possessing the

lowest bandwidth. For example, if there are two distinct links between a workstation and a server and one link has a bandwidth of 1.544 Mbps and the other link has a bandwidth of 44.736 Mbps, the effective throughput will not exceed 1.544 Mbps. Another factor is the time it takes for a given server to process a request. Issues such as queue-length, processing speeds, request size, processing overhead, and background processing impact the length of time it takes a server to respond to a given request. Most of these issues are dealt with by the OPNET server model and do not require any special settings.

There are however, some server processing issues that are not automatically handled by the model and were not considered in this research. Specifically, server initialization time, database access times, and background processing were not factored in since these issues do not change the results when comparing the throughput of one scenario with another. Even if these issues were factored in, the throughput values obtained in each simulation would change by the same amount so no benefit is gained by considering them. On the other hand, these issues would be important to consider if the intent was to discover what impact they would have on the throughput of an individual system. Since this is not the goal, server initialization times, database access times, and background processing are not factored in. Throughput values for each scenario are measured and compared to determine which scenario provides the best throughput.

3.8.2 Application response time

Response time is defined as the time between the end of the user's request (i.e., when the user has finished submitting the request) and the time the system has completed its response to the user. Response time is a lower-better metric and directly impacts throughput. For example, the longer it takes a server to process a job, the lower the

throughput will be. For the simulations being run in this research effort, application response time is the metric of primary concern. A custom application has been defined in OPNET that emulates a user logging onto the VDL, submitting a query, then requesting to download ATR image files based upon the results of the query. Once the last requested image file has been received by the user, the application ends and the elapsed time is recorded as "application response time." This is the response time that will be used to compare the performance of the three scenarios being modeled. The scenario with the best application response time will be deemed the best performer. All other server functions are being simulated in OPNET's server model.

3.9 Parameters

Simulating a network in OPNET requires the setting of many network component-specific parameters. Each node (server/workstation) or link (ATM) in the model has multiple attributes that require specific values. For this research effort, to simplify matters, unless specifically required for purposes of modeling the VDL, all attributes are left with their "default" values unchanged, except as noted elsewhere. Only those attributes requiring VDL-related values are discussed in this section. All remaining attributes and their values are detailed in Chapter 4 along with the specific values used for the parameters discussed in this section. The following list contains those parameters that require specific values for the purpose of modeling the VDL. Following the list are brief descriptions of the parameters that are integral to the VDL simulation.

System parameters:

- network bandwidth,
- connection speeds,

- service times, and
- link background utilization.

Workload parameters:

- file sizes,
- number of files being requested, and
- number of users.

3.9.1 Network bandwidth

Network bandwidth is an obvious system parameter of interest since bandwidth directly impacts throughput. The higher the bandwidth, the more data can be transmitted through the medium. Of course, there are other factors influencing throughput such as server response times, lost packet recovery (as with TCP/IP), however, the highest bandwidth obtainable is the primary limiting factor in any network. For example, regardless of how fast a server can process requests, throughput is limited to the bits/second that can be transmitted across the medium. If a server can process jobs faster than they can be transmitted over the medium, the server will have to compensate by queuing the outgoing jobs which can ultimately result in reduced throughput. A similar situation can occur if the bandwidth is such that data arrives at a faster rate than the server can process it. Once again the server will have to compensate by queuing the incoming requests. In each example, bandwidth directly impacts throughput. This demonstrates the need to understand how making changes in a network, whether it is increasing bandwidth or upgrading a server, can impact overall performance. Given the possibilities, the bandwidth of selected portions of the VDL network model is varied

during the simulations in order to determine the impact these variances will have on overall throughput and application response time.

3.9.2 Connection speeds

As discussed in section 3.5, user and central server connection speeds were varied to determine the impact on application response time.

3.9.3 Service times

Server response times must be calculated to validate results obtained through OPNET simulations. Since *application response time* is the primary metric of concern, it must be mathematically calculated to validate the results obtained. To accomplish this, service times, file sizes, and link data rates (bandwidths) must be known. Once service times have been calculated, they can be used in conjunction with file sizes and the associated link data rates to calculate the expected application response time.

3.9.4 Link background utilization

To add more realism to the simulations, the network is assumed to be lightly loaded and a 10% link background utilization is factored into the simulations. This background traffic represents users performing other tasks on the network (e.g., e-mail and http) non-related to the downloading of ATR image files.

3.9.5 File size

File size is an important workload parameter. In this research effort, it is also a factor therefore any discussion on file size is deferred to the next section.

3.10 Factors

The factors used for this research effort are *file size*, *connection speeds, traffic patterns (scenarios),* and *number of users*. The following sections provide more detail on

file size, *connection speeds*, and *number of users*. Scenarios were discussed in the sections 3.2 through 3.4.

3.10.1 File sizes

In addition to being an important workload parameter, file size is also a factor impacting system performance. One of the primary tasks of this research is to demonstrate the impact different file sizes have on the throughput and application response time of the system. Discussions held with VDL designers and ATR researchers regarding file sizes reveal that there are no typical file sizes for ATR images, algorithms, or result sets (output). ATR Images can range in size from a 10KB (chip image) to a 1GB hyperspectral image. Files containing algorithms and their associated databases, structures, and templates, may range in size from 500KB to 3GB. Likewise, the output returned to the user can range from 10KB to 10MB depending upon the level of detail the user wants included in the output. For these reasons, files sizes were selected based upon the best predictions and estimates provided by ATR researchers and VDL designers [BAE00].

VDL designers predict users may require up to 3,000 or more image files for processing by a single algorithm. In an effort to reduce network traffic, designers have decided image files requested by a user will be consolidated into compressed files for transfer. While increasing the total file size (for example a single 1MB file as opposed to ten 100KB files), compressed files will reduce the total number of files being transferred over the network. VDL designers are leaning towards three sizes for the compressed files. The sizes are 1MB, 10MB, and 100MB. During the simulations, these file sizes will only be used in scenarios 1 and 3 (Figures 6 and 8) since they are the only scenarios

where image files are being transferred over the network. In scenario 2 (see Figure 7), only algorithm files and result sets (output files) are being transferred. The file sizes which are used for simulating scenario 2 (centralized processing) are 1GB for the algorithm file and 1MB for the output file. As for input file sizes, previous VDL demonstrations have shown that user image queries were no larger than 10KB in size. Since this value is also sufficiently large enough to contain the necessary overhead (bytes for destination address, source address, preamble, etc.) associated with a request to download a file, 10KB is used in the simulations to represent all requests [BAE00]. Furthermore, it is assumed all file sizes vary according to a normal distribution.

3.10.2 Connection speeds

Connection speeds (bandwidth) are varied in order to demonstrate the performance advantages (increase in throughput and application response time) obtained with higher bandwidths. Intuitively, higher connection speeds usually result in higher throughput, however the magnitude of performance improvement is what is of major interest. The simulations demonstrate the improvement in throughput and application response time as the result of higher connection speeds. This information may prove useful in justifying the additional costs associated with greater bandwidth. Additionally, the results of the simulations may provide ammunition for obtaining approval for a dedicated OC12 link between the VDL central server and the DREN. During the simulations, user connection speeds are varied between T1 and T3 (this value is set to 8Mbps for those experiments where the central server connection speed is set to 8Mbps). The connection speed between the central server and the DREN is varied between 8Mbps (current capability) and OC-12 (desired capability). Table 1 lists the factors.

3.10.3 Number of users

Approximately 200 users are currently slated to participate in the VDL [BAE00]. Given the potential for multiple users accessing the VDL simultaneously, this was considered an important factor to consider in the simulations. Assuming no more than 10% of the users attempt to access the VDL at the same time, the number of users is varied between 2, 10, and 20 during the simulations. The resulting data demonstrates to designers of the VDL the impact simultaneous access has on application response time and throughput for the competing scenarios.

Table 1. Factors

Scenarios	File Sizes	Connection Data Rates	Number of Users	Central Server Connection Speeds
1	1Mbyte	T1	2	8Mbps
2	10Mbytes	*T3	10	OC12
3	100MBytes	N/A	20	N/A

3.11 Evaluation Technique

The evaluation technique used for this research is *simulation*. Table 2, taken from [JAI91], illustrates the advantages and disadvantages of the three different evaluation techniques. Simulation was selected for two main reasons. The primary reason is that given there is no operational system (the VDL has not been fully implemented) from which measurements can be obtained, simulations are required to estimate performance statistics. Additionally, simulations have a higher degree of credibility in the eyes of the customer as opposed to analytical models, which can only provide trend data as opposed to more realistic performance data. As shown in Table 2 adapted from [JAI91],

analytical models have a low level of accuracy. With the results described in Chapter 4, designers of the VDL can more accurately predict the way the VDL will perform based upon certain factors enabling them to make more informed decisions.

Table 2. Criteria for selecting an evaluation technique

Criterion	Analytical Modeling	Simulation	Measurement
Stage	any	any	postprototype
Time required	small	medium	varies
Tools	analysts	computer languages	instrumentation
Accuracy	low	moderate	high
Trade-off evaluation	easy	moderate	difficult
Cost	small	medium	high
Saleability	low	medium	high

3.12 Workload

The workload selected for this research has multiple aspects. The workload consists of a user download request, a specific traffic pattern (scenario), and a specified number of users. Traffic over the network varied in size (bytes) and routing (direct download versus download via the central server) depending upon the scenario simulated. Each user request places a demand on the system that results in data files being transmitted over the network. The factors shown in Table 1 characterize user requests. For example, assuming a user requests three thousand 100KB files, and 10MB compressed files are being used, this would require the transmission of thirty 10MB files (as was the case in the simulations). This will place a different load on the system than a

request for thirty 100MB files. Additionally, depending on the scenario simulated, different loads are placed on the central server as well as the network itself. Finally, the number of users requesting to download files also changes the load on the system thus impacting performance. For example, ten users simultaneously downloading files will place more of a demand on the system than a single user. For this reason, the number of users requesting downloads is varied during the simulations. Since VDL designers are currently aware of approximately 200 potential users of the system, 20 was selected as the maximum number of users for simulation purposes under the assumption no more than 10% of the total number of participants will attempt to download files at any given time. Given this maximum value, the number of users is varied between 2, 10, and 20 during the simulations.

3.13 Experimental Design

The experimental design applied in this research effort is the full-factorial design with replications. This design was selected since each factor is believed to have the potential of significantly impacting system throughput and application response time. Additionally, replications were used so experimental error could be factored in for more accurate results. Utilizing the above design, the factors from Table 1, and the fact that five replications per experiment were run, the total number of experiments conducted for scenarios one and three is:

2 x 3 x 2 x 3 x 2 x 5 = 360 experiments

Scenario two (centralized processing) is slightly different since the file sizes are held constant (the size of the algorithm file and results file do not change from one experiment

to the next). For this reason, fewer experiments are conducted for this scenario. The number of experiments required is:

2x3x2x5=60 experiments

This brings the total number of experiments conducted to **420**. Five replications were run so accurate standard deviations and variances could be obtained and experimental error could be factored into the results. Based upon the results of the simulations, the values obtained for throughput and application response time are statistically compared to determine if one scenario is significantly different from another and at what level of confidence.

3.14 Summary

The methodology introduced in this chapter outlines how different collaboration scenarios were developed and what parameters and factors were important to the experiments. Additionally, the evaluation techniques and type of experimental design were identified. Recapping, the steps followed were:

- define problem – Simulate various collaboration scenarios to determine which scenario provides the best overall performance (throughput and application response time).
- define system boundaries – The system boundaries consisted of the system under test (SUT) and the component under test (CUT). The SUT consists of servers, workstations, databases, interconnection network, and Major Shared Resource Centers (MSRCs). The CUT for this research was the interconnection network.

- <u>list system services</u> – Distributed access to large data repositories, distributed access to powerful computing resources, and high-bandwidth capability for transmission of large amounts of data.
- <u>list performance metrics</u> – Application response time and throughput.
- <u>list parameters (system and workload)</u> – Network bandwidth, connection speeds, service times, link background utilization, file sizes, and number of files requested.
- <u>identify factors</u> – File sizes, scenarios, user connection speeds, central server connection speed, and number of users.
- <u>identify evaluation technique</u> – Simulation.
- <u>select workload</u> – User download request, traffic pattern (scenario), and number of users.
- <u>choose experimental design</u> – Full-factorial design.

Following this methodology, the results obtained will provided statistical insight into the kind of performance that can be expected from the different collaboration scenarios evaluated. Utilizing this information, more informed design decisions regarding the ultimate implementation of the VDL can be made.

4. Implementation and Analysis

4.1 Introduction

This chapter discusses how the three collaboration scenarios described in Chapter 3 were implemented in a simulation environment. The results obtained from simulating these scenarios are provided with analysis. Section 4.2 briefly introduces OPNET Modeler, the modeling tool used for the simulations and provides implementation details for each of the components (nodes) used in the network models. The components discussed are the workstation node, server node, ATM switch node, ATM link node, ATM cloud node, task configuration utility object, application configuration utility object, profile configuration utility object, permanent virtual circuit configuration utility object, and the simulation configuration object. Section 4.3 discusses the results of the simulations and section 4.4 summarizes the results. In short, results showed that increasing the central server connection bandwidth from 8Mbps to 622.08Mbps resulted in modest or negligible performance gains when users were limited to the lower bandwidth range of 1.544 - 44.736Mbps. Additionally, it was determined there was no difference in performance between scenarios 1 and 3. Either of these scenarios provides better application response time if the total amount of data required by the user is less than the size of the algorithm file and result file combined. Otherwise scenario two (centralized processing) provides better application response time.

4.2 OPNET Modeler

OPNET Modeler is a simulation program for networks. Modeler can incorporate proposed changes and determine how the network will perform. For example, consider

an organization interested in upgrading a router or some other component(s) in a network. The organization wants to ensure the upgrade cost will be justified in light of the performance improvement. OPNET can model the network with several different routers. Simulations can then be used to compare the performance of the network using the different routers. Once performance statistics have been gathered, a performance-cost analysis can be conducted to choose a router. OPNET can also be used to simulate the behavior of routing algorithms, different network topologies, and proposed configurations. The network or system under test for this research consists of workstations, servers, ATM switches, an "ATM cloud," and ATM links. Additionally, a permanent virtual circuit (PVC) configuration utility, task configuration utility, application configuration utility, and profile configuration utility objects were used to simulate the data traffic patterns associated with the different scenarios. Descriptions of each of these components and how they were configured are discussed below.

4.2.1 Workstation implementation

The workstation node used in the network models is the "atm_wkstn_adv" (advanced ATM workstation) node. This node is used to represent an ATM node with client-server applications running over TCP/UDP [OPN00]. Table 3 lists those attributes of the "atm_wkstn_adv" node that required modification from their default values. The default values were sufficient for all other attributes and therefore are not listed. The first column contains the name of the attribute, the second column contains the attribute's value or setting, and the third column contains the higher-level attribute(s) that must be accessed in order to reach the attribute listed in the first column. For example, if the attribute in question is four levels deep, three attribute names will appear

Table 3. Workstation node attributes

Attribute Name	Value/Setting	Access Tree
Peak Cell Rate (PCR) in Mbps	622Mbps	1. ATM port buffer configuration 2. Traffic Parameters (UBR)
Minimum Cell Rate (MCR) in Mbps	622Mbps	1. ATM port buffer configuration 2. Traffic Parameters (UBR)
Sustainable Cell Rate (SCR) in Mbps	622Mbps	1. ATM port buffer configuration 2. Traffic Parameters (UBR)

in the third column with the first name representing the highest-level attribute (starting point) and proceeding on in descending order. If the column is marked "N/A," the attribute is not a lower level attribute. This table format is adhered to throughout this chapter.

In addition to the changes made to the attributes listed in Table 3, the atm_wkstn_adv node was configured to use OPNET's custom applications. When using a custom application, sources and destinations are specified at both the workstation and server nodes in the model. Figure 12 shows the application destination preference table

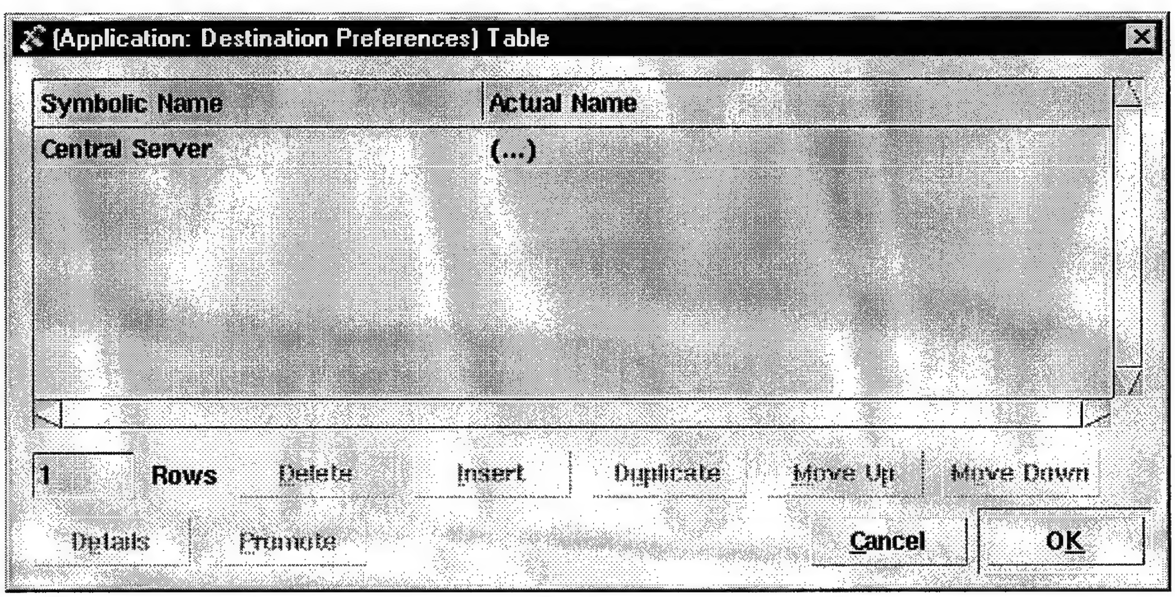
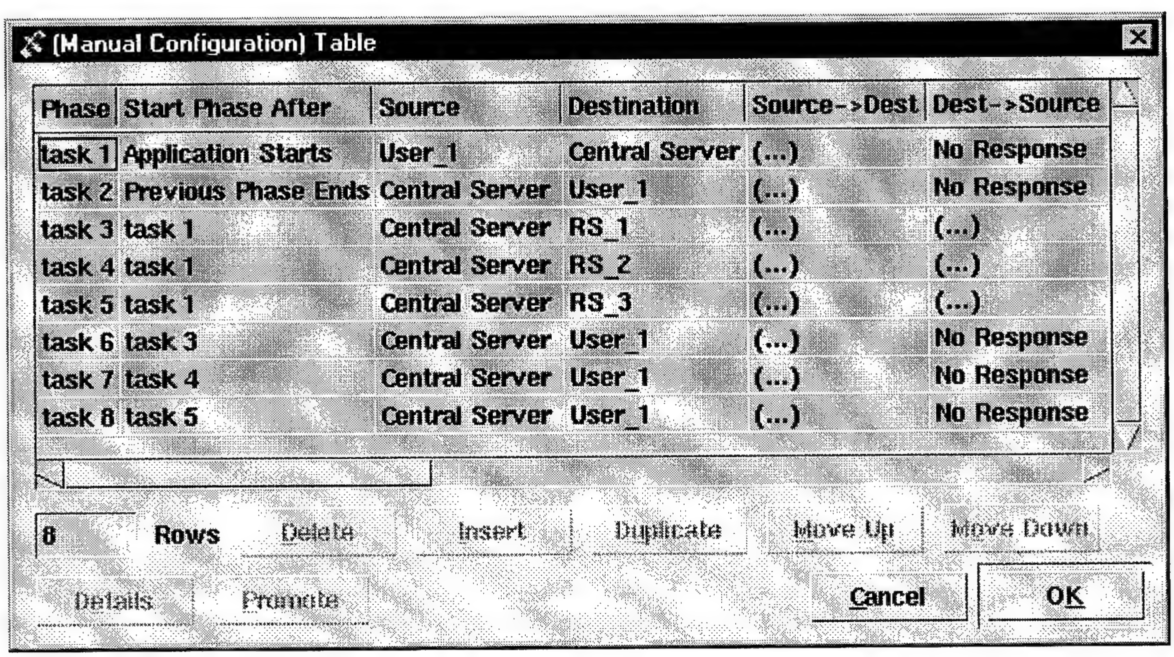


Figure 12. Application destination preference for workstation node

for the workstation node. The symbolic name “Central Server” identifies a specific server in the model, in this case, the central server. This symbolic name must match the symbolic name used in the application manual configuration table (shown in Figure 13). In Figure 13, the first application phase (task 1) shows communication between User_1 and the Central Server. User_1 is the symbolic name of a workstation node and Central Server is the destination preference (Figure 12). The symbolic name for the central server must be the same in both tables for proper operation of the application. This applies to all symbolic names in the model (workstation and server nodes). If the names do not match, the application will fail. Another important aspect of application destination preferences is the “actual name” attribute, also shown in Figure 12. The name specified in this attribute (not shown in figure) must match the name specified in the “server address” attribute of the destination (server) node. Again, if these names do not match, the application will fail because the symbolic name is mapped to the address of



Phase	Start Phase After	Source	Destination	Source->Dest	Dest->Source
task 1	Application Starts	User_1	Central Server	(...)	No Response
task 2	Previous Phase Ends	Central Server	User_1	(...)	No Response
task 3	task 1	Central Server	RS_1	(...)	(...)
task 4	task 1	Central Server	RS_2	(...)	(...)
task 5	task 1	Central Server	RS_3	(...)	(...)
task 6	task 3	Central Server	User_1	(...)	No Response
task 7	task 4	Central Server	User_1	(...)	No Response
task 8	task 5	Central Server	User_1	(...)	No Response

Figure 13. Manual configuration (phase) table

the destination (server) node, which happens to be the "server address" attribute of the node. Also of extreme importance are the "application source preferences" and "application supported profiles" attributes. "Application source preferences" is the symbolic name of the workstation node itself and is how the node is identified in the manual configuration table (node's actual address). For example, in Figure 13, task 1, a source node has been identified as "User_1." This indicates there is a workstation node with its "application source preference" set to "User_1" as shown in Figure 14. The

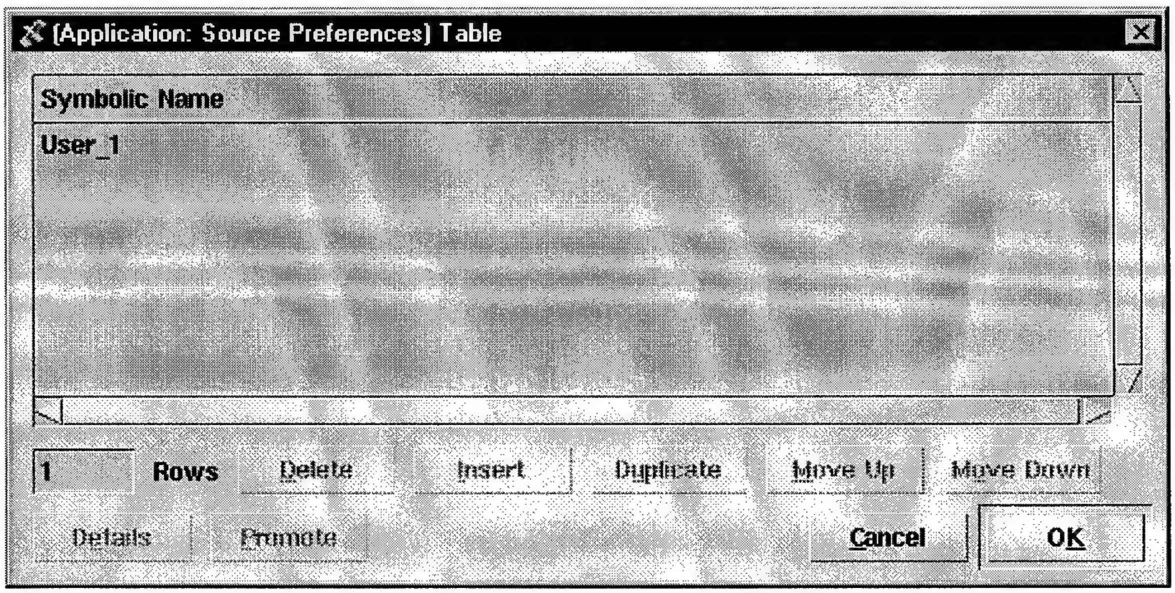


Figure 14. Application source preference for workstation node "User_1"

"application supported profiles" attribute must contain the name of a profile that was created and exists in the profile configuration utility object (discussed later). Briefly, a profile is used to describe a particular user and to generate application layer traffic [OPN00]. Workstation and server nodes can support different profiles allowing for more flexibility in the simulations. It is important to know that if any of the attributes previously discussed are left blank; the custom application will not work. The only exception to this is the "application supported profiles" attribute. In the version of

OPNET used for this research, a bug exists preventing the use of this attribute along with the "application supported services" attribute (this does not affect the workstation node, but it does affect the server node discussed in the next section). Fortunately, only the "application supported services" attribute was required and therefore this bug did not impact the results of this research. Prior to any future research, however, it would be advisable to have the most current version of OPNET installed to avoid any potential problems.

4.2.2 Server implementation

The server node used in the simulation models is the "atm_server_adv" node. This server node represents an ATM node with client-server applications running over TCP/UDP [OPN00]. As was the case with the workstation node, the advanced server node must be used since use of the custom application feature was required in order to simulate the three competing scenarios. Server node attributes were left at their default settings except for the processing speed multiplier attribute of the central server node, which was set to "2," and the PCR, MCR, and SCR attributes which were identical to the values shown in Table 3. The processing speed multiplier of the server node was set to "2" since the central server currently in use by the sponsor is a dual processor machine. Additionally, it was assumed the central server was twice as fast as any remote server was. Furthermore, in the absence of a VDL specification, certain assumptions were required.

Custom application-related attributes for the server such as "application destination preferences" and "application source preferences" are set up in the exact same way as they were with the workstation node. The only difference is the existence of a

"application supported services" attribute. The "application supported services" attribute is used to define what applications the server will run.

4.2.3 ATM switch implementation

Five switches were used in the network models. The switch node used is the atm8_crossconn_adv node model. This model implements VP and VC switching capabilities in an ATM network [OPN00]. The switches represent the points in the network where workstations and servers connect to the Defense Research and Engineering Network (DREN). The only switch attributes modified were the PCR, MCR, and SCR attributes (see Table 3 for the values used).

4.2.4 ATM link implementation

The ATM_adv link node was used to connect ATM switches, gateways, and station nodes at selectable data rates [OPN00]. Three attributes of the link node were modified for the simulations. Table 4 lists the attributes and their values. The *data rate* attribute has four values listed because data rate was one of the factors varied from experiment to experiment. The values shown represent the levels used in the experiments. *Background link utilization* was set to 10% to simulate a lightly loaded network. Since propagation delay was not factored into the results, *delay* was set to zero and *propagation speed* was set to "speed of light."

Table 4. ATM_adv link node attributes

Attribute Name	Value/Setting	Access Tree
propagation speed	speed of light	N/A
background utilization (%)	10	1. background utilization
data rate	T1, T3, 8Mbps, OC12	N/A
delay	0	N/A

4.2.5 ATM cloud implementation

The ATM cloud node, ATM32_cloud_adv node, represents an ATM cloud through which traffic is modeled using 32 input/output physical links [OPN00]. As was the case with the ATM switch, the only attributes of the ATM cloud requiring modification were the PCR, MCR, and SCR attributes. The attribute settings used are listed in Table 3. Once again, the values selected were based upon the desire of the VDL designers to achieve OC12 data rates over the network.

4.2.6 Custom application implementation

The custom application feature of OPNET was used to model specific data traffic patterns. For example, in the baseline scenario all automatic target recognition (ATR) images are processed through the central server whereas in the direct download scenario (scenario 3), users download desired ATR images directly from the remote server(s). Setting up a custom application requires the configuration of multiple configuration utility objects. Each of these utility objects works in conjunction with each other and the "application source preferences," "application destination preferences," "application supported services," and the "application supported profiles" attributes of the workstation and server nodes in the model. The configuration utility objects required for the models used in the simulations are the task configuration utility object, application configuration utility object, profile configuration utility object, and the permanent virtual circuit (PVC) configuration utility object. Each of these utility objects is explained in detail in the next four sections.

4.2.6.1 Task configuration utility object

The task configuration utility object is used to create tasks that characterize a custom application. Traffic patterns, file sizes, and request and response times are defined here. Once these tasks are created, applications may be defined that utilize these tasks which are in-turn used to create a user profile. The user profile is specified at selected nodes for the purpose of characterizing the traffic processed by that node [OPN00]. Figure 15 shows the top-level attributes of the task configuration utility object. To access the *task specification table* where tasks are created and identified for use, the *task*

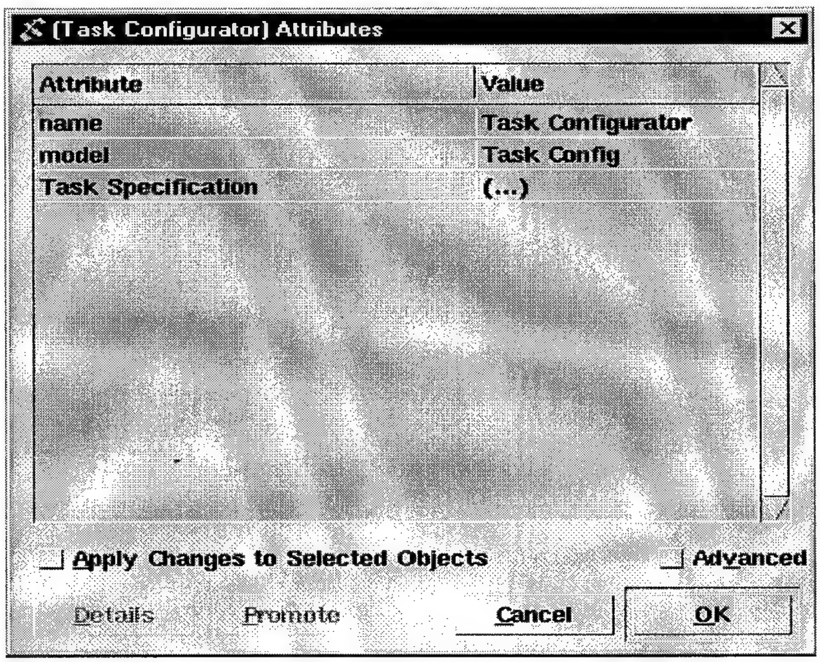


Figure 15. Task configuration utility attributes

specification attribute must be edited. Once inside this attribute, the *task specification table* shown in Figure 16 may be edited. Upon accessing the *task specification table*, desired tasks can be created by naming them and configuring them through the *manual configuration* attribute which brings up the *manual configuration table* (Figure 13). The

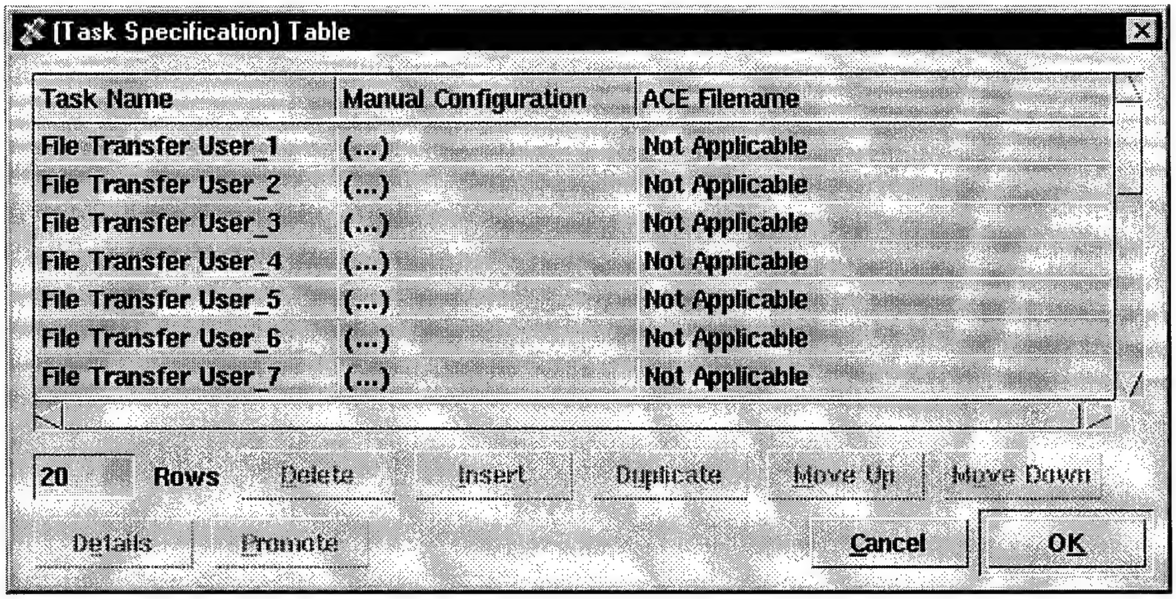


Figure 16. Task specification table

manual configuration table is where data traffic patterns are created which represent the different scenarios that were simulated. In Figure 13, eight tasks represent the baseline scenario of a user submitting a download request to a central server which then fulfills the request or forwards the request to remote servers to obtain those images it does not have locally. The user sends a request to the central server (task 1) to download thirty image files. The central server has access to twenty of the requested image files so it immediately starts sending them back to the user (task 2). The other ten files must be retrieved from remote servers so the central server forwards requests to the appropriate remote servers for retrieval (tasks 3, 4, and 5). Five files come from remote server 1 (RS_1), three from RS_2, and two from RS_3. Once the central server starts receiving the requested files from the remote servers, it starts forwarding them to the user who requested them (tasks 6, 7, and 8). Not shown in Figure 13 are the "request/response pattern," "end phase when," and "transport connection" attributes. The "request/response pattern" attribute determines whether or not requests and responses occur serially or

sequentially. The "end phase when" attribute is used to specify when a phase is considered completed. For example, if "when final response arrives at source" is selected, the phase will not end until the final image file has been received by the source (requestor). The "transport connection" attribute is used to specify whether or not the same connection will be used for all data transfers that occur within a phase. It is important to note that when modeling a network where servers are receiving and responding to requests at the same time (concurrent transactions occur), the "transport connection" must be set to "new connection per request." Otherwise, the application will not function properly. The "start phase after" attribute also needs to be discussed. This attribute is used to specify when each phase in the table starts. If set to "application starts," the phase will begin as soon as the application begins. If set to "previous phase ends," the phase will not start until the preceding phase has completed. Another option is to enter in a specific phase name. For example, in Figure 13, the sixth phase in the table, identified as "task 6" will not start executing until task 3 has completed. If a phase must wait for multiple other phases to complete, then a comma-separated list of phase names may be entered in which tells the application to wait for these particular phases to end before execution of this phase begins.

Table 5 lists the attributes of the task configuration utility object that were modified for the experiments. While the *manual configuration table* shown in Figure 13 will look different for scenarios two and three (see Appendix D), the rest of the attribute settings for the task configuration utility object will for the most part be the same. The only differences are the file sizes used in scenario two (centralized processing). In Table 5, the file size with the number two in parenthesis next to it was used in scenario two

only. Additionally, those settings containing multiple file sizes because the file size was varied between simulations. Otherwise, the attributes listed apply to all three scenarios.

4.2.6.2 Application configuration utility object

The application configuration utility object is used to select applications that characterize the type of data traffic occurring over a network. For example, http, ftp, voice, and video are some application options that may be selected. Additionally, if a

Table 5. Task configuration utility object attributes

Attribute Name	Value/Setting	Access Tree
initialization time (seconds)	constant (0)	1. task specification 2. manual configuration 3. source->dest traffic
request count	constant (1)	1. task specification 2. manual configuration 3. source->dest traffic
inter-request time (seconds)	constant (1)	1. task specification 2. manual configuration 3. source->dest traffic
request packet size (bytes)	constant (10,000/1,000,000/ 10,000,000/100,000,000) (2) 1,000,000,000	1. task specification 2. manual configuration 3. source->dest traffic
packets per request	constant (1)	1. task specification 2. manual configuration 3. source->dest traffic
inter-response time (seconds)	constant (1)	1. task specification 2. manual configuration 3. dest->source traffic
response packet size (bytes)	constant (1,000,000/ 10,000,000/100,000,000)	1. task specification 2. manual configuration 3. dest->source traffic
packets per response	constant (1)	1. task specification 2. manual configuration 3. dest->source traffic
policy	new connection per request	1. transport connection

custom application has been created, it may also be selected using this utility object. Figure 17 shows the attributes of the application configuration utility object. The "application definitions" attribute is the attribute of primary concern since this is where all modifications to this utility object occur. When editing this attribute, the *application*

definitions table window pops up as shown in Figure 18. In this table, applications are named and the type of application simulated is selected. In Figure 18, the first application in the table is called, i.e., “ATR Image Retrieval_User_1.” Accessing the details of this application requires the editing of the “description” attribute. Figure 19 shows the window that pops up when this attribute is selected. Any one of the

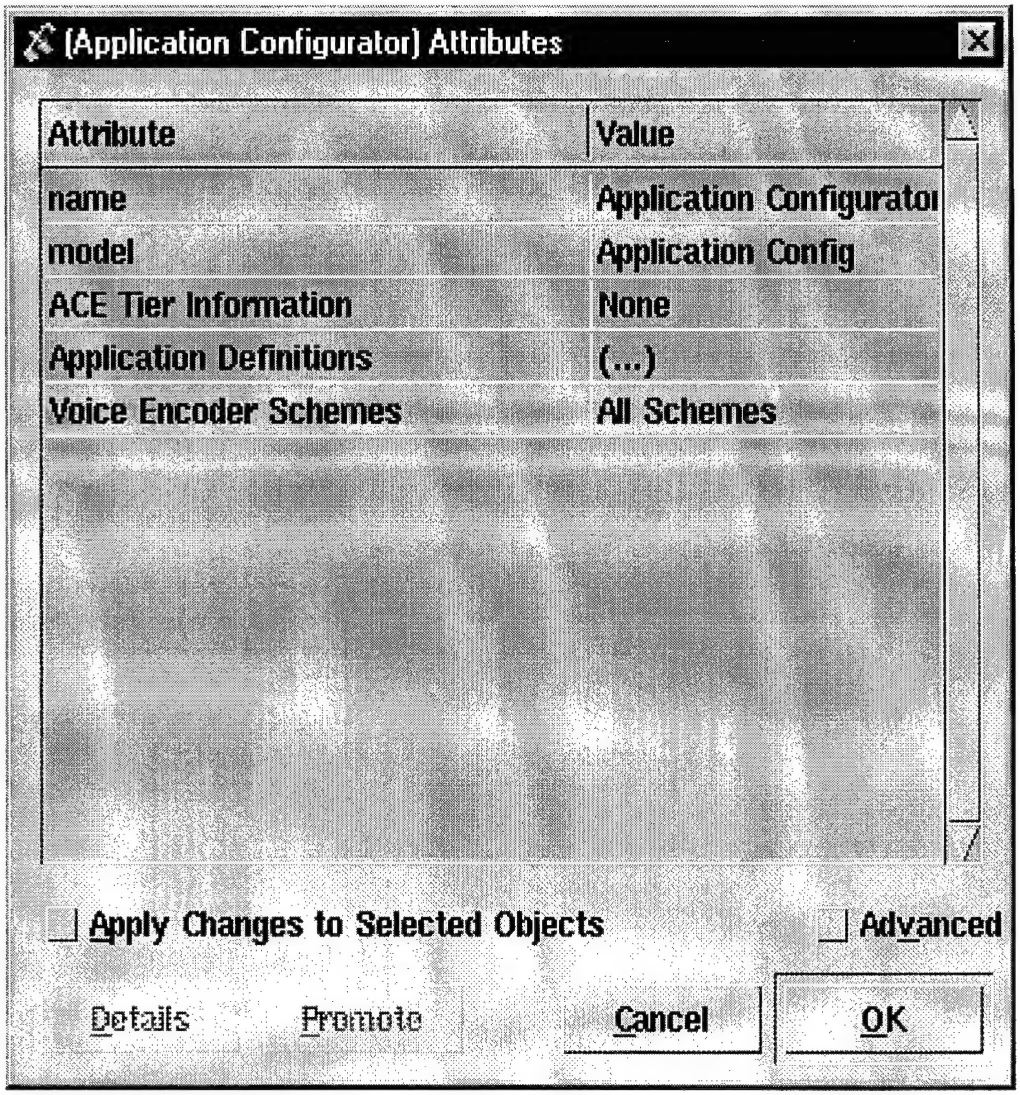


Figure 17. Application configuration utility object attributes

applications listed in this window may be selected if so desired, however, for this research a custom application was created. The custom application is selected by editing

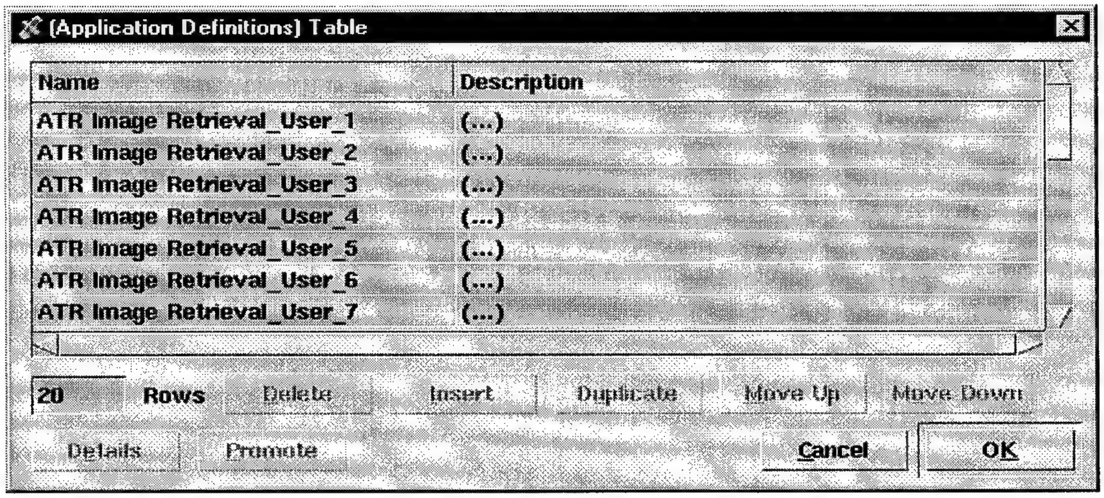


Figure 18. Application definitions table

the "custom" attribute. This brings up the window shown in Figure 20. Here, several application-specifics may be modified such as the transport protocol. For this research effort, the default values were used. The "task description" attribute is the next attribute that must be accessed. This attribute is where specific tasks are identified. The task configuration utility object is where tasks are created for the purpose of generating specific amounts and pattern of data traffic across a network. Editing this attribute is how

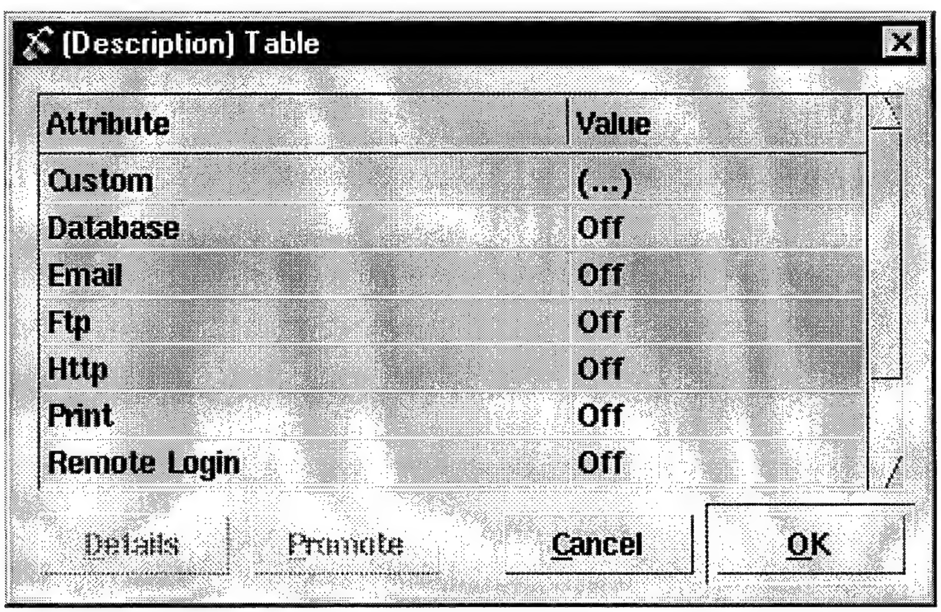


Figure 19. Application description table

those tasks are selected as part of the application that will run during the simulations. Since the only tasks selectable are those that were created using the task configuration

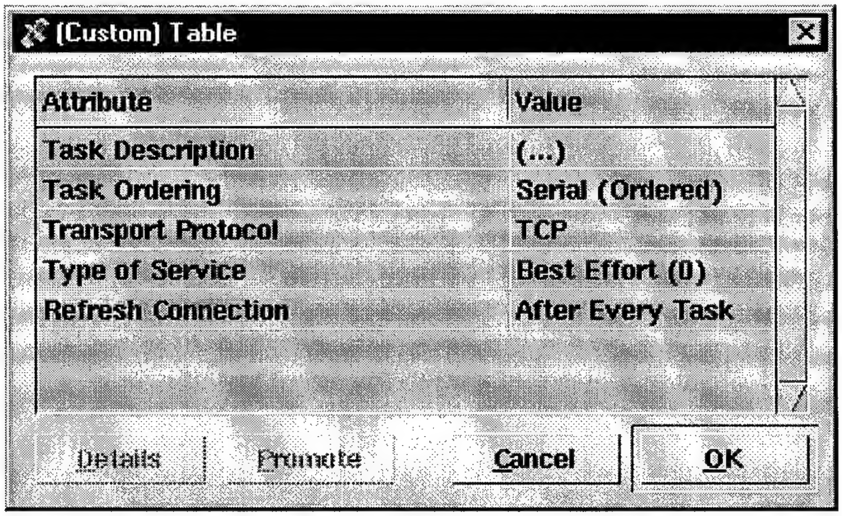


Figure 20. Custom application description table

utility object, tasks must be defined before configuring the application. Figure 21 shows the *task description table.* In this example, the task selected was a task previously created called "File Transfer User_1." Although only one task is listed, more than one task may be selected for a given application. The other attributes of the task description

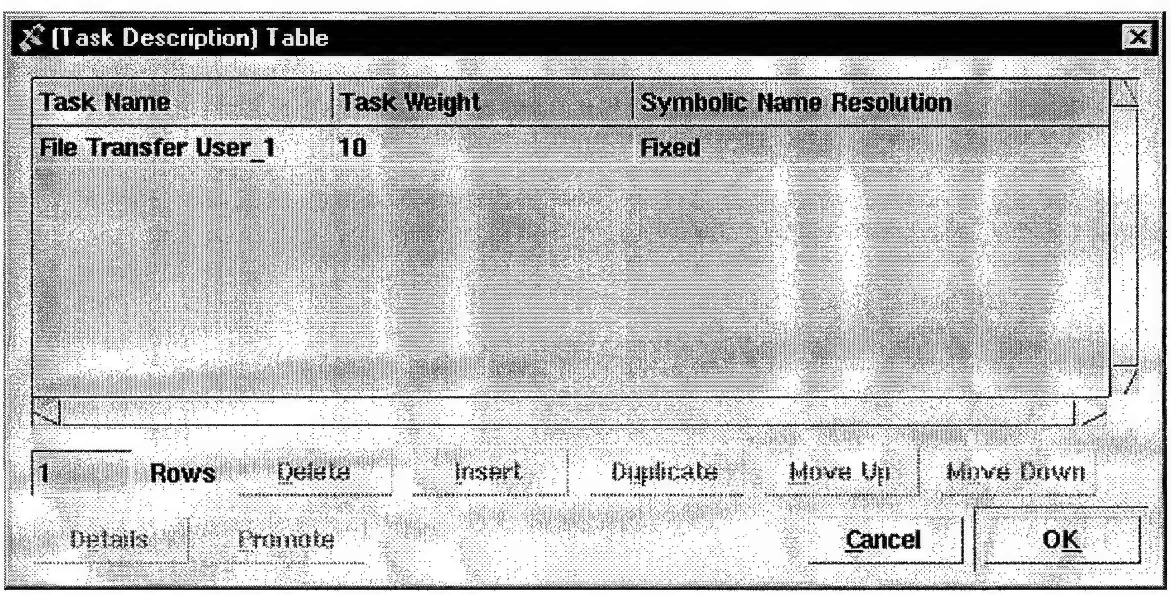


Figure 21. Task description table

table were not changed. The task weight is only used in situations where the custom application is not used and a weighting scheme is required to determine what percentage of traffic is simulated as one type of application versus another.

4.2.6.3 Profile configuration utility object

The profile configuration utility object is used to create user profiles. User profiles characterize the network usage of a specific user on the network. One profile might represent a user who the majority of the time browses the internet (an http application), while another profile represents a user that uses the ftp application. User profiles can be specified on different network nodes for the purpose of generating application layer traffic [OPN00]. Figure 22 shows a profile configuration table where user profiles are specified. The attributes of this table are *profile name*, *applications*, *operation mode*, *start time*, and *duration*. The *profile name* attribute is

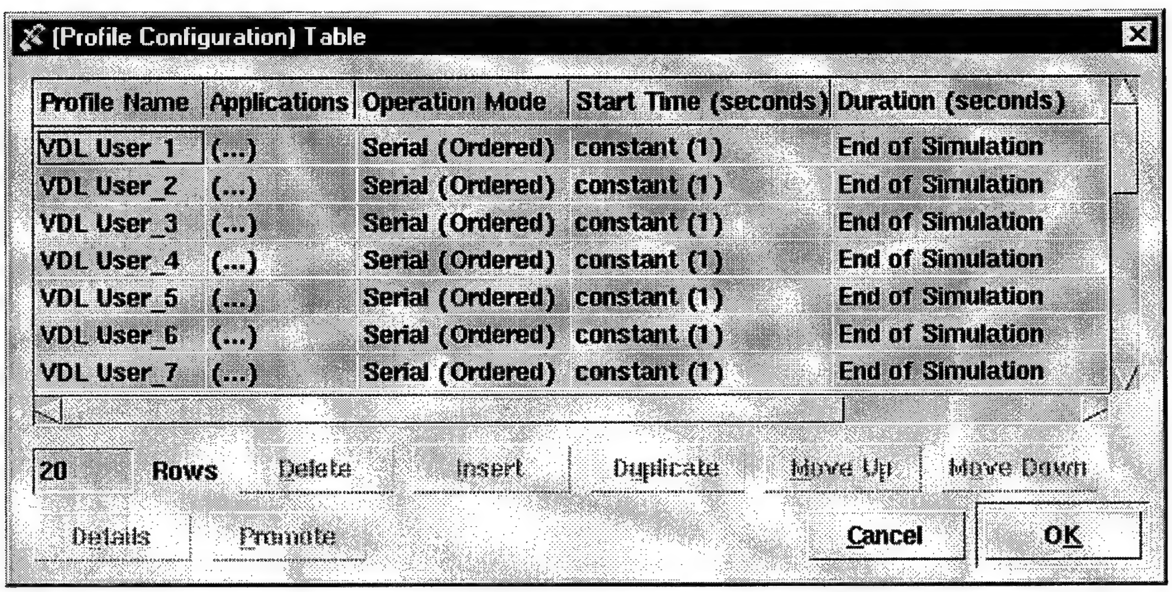


Figure 22. Profile configuration table

where specific user profiles are identified. Using the previous examples, a user profile could be called "http user" or "ftp user." The profile names shown in the figure were

used for this research. The *applications* attribute is accessed to specify applications that a user uses, such as http or ftp. When editing this attribute, an application table pops up in a window as shown in Figure 23. The attributes of this table are *name*, *start time offset*, *duration*, and *repeatability*. The *applications table-name* attribute contains the names of the applications that characterize the user being profiled. When editing the name, the only options available will be those applications that were created using the application configuration utility object. So, applications need to be defined prior to configuring user profiles. The *applications table-start time offset* attribute only applies if more than one application is selected for a user profile. This offset normally refers to the time between the end of one application and the start of the next when applications are set up to run serially. If the applications are configured to run simultaneously, then this time refers to

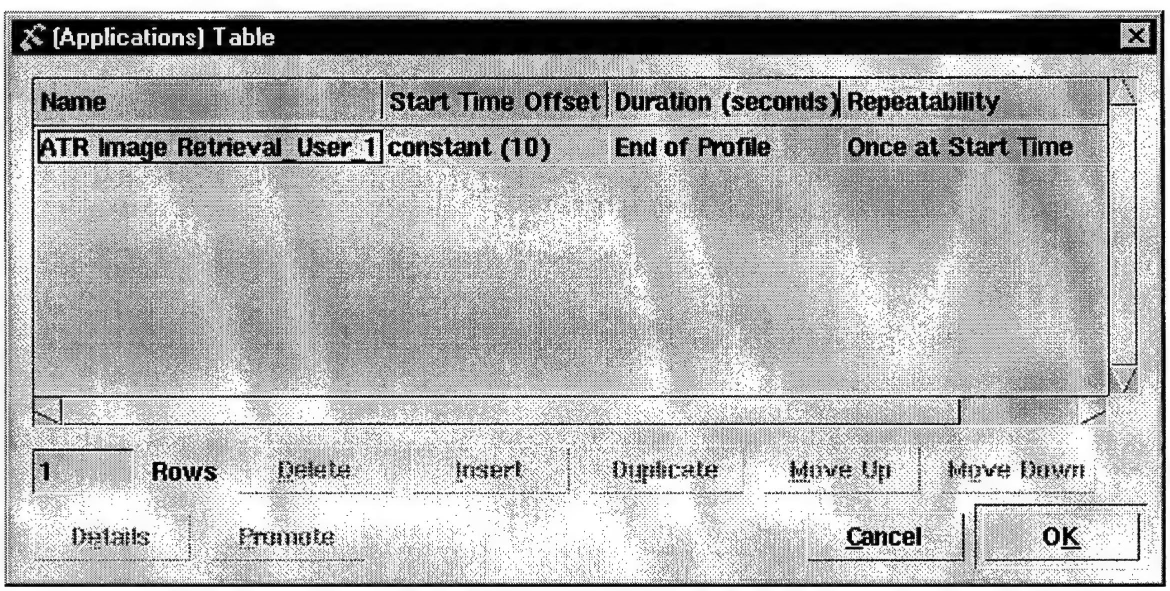


Figure 23. Applications table

the time between the start of the user profile and when the application will start. This attribute was not applicable to this research. The *applications table-duration* attribute identifies how long the application will run. The *applications table-repeatability*

attribute identifies how many times within a profile the application will repeat. If you want an application to continually run for the duration of the profile, then this attribute can be set to "unlimited." In Figure 22, the *operation mode* attribute identifies how the applications shown in Figure 23 will execute. They can either execute serially (as was the case for these experiments) or they can execute simultaneously. The *start time* specifies the start time of the profile. As an example, if traffic on a network was particularly bursty, say, high usage early in the morning and then again at midday, the user profiles can be configured to start at different times to allow for the simulation of this type of network usage. Finally, the *duration* attribute identifies how long the user profile will run. The settings shown in Figures 23 and 24 were used in all experiments conducted for this research.

4.2.6.4 Permanent virtual circuit configuration utility object

The permanent virtual circuit configuration utility object is used to define permanent virtual circuit (PVC) configurations. Depending upon the scenario, PVCs are established between user workstations and the central server, between the central server and remote servers, and between the users and the remote servers. Figure 24 shows the PVC configuration used for two users connected to the DREN via T1 connections. The central server in this experiment connects to the DREN via an 8Mbps connection. The "source" attribute is the symbolic name of the node as is the "destination" attribute. The "traffic contract attribute" is accessed in order to specify the requested data rate of the PVC. It is important to note that if the requested data rate is greater than the supported data rate of any of the links in the PVC, the application will fail. That is, the requested data rate cannot exceed the data rate of the slowest link in the PVC.

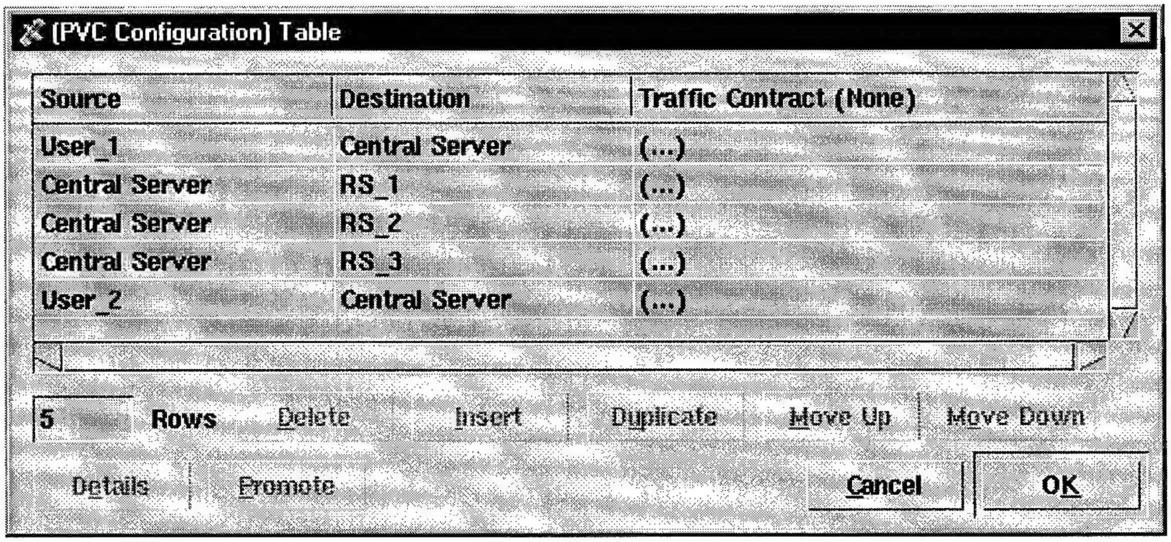


Figure 24. PVC configuration table.

When editing the "traffic contract" attribute, the first window to pop up contains the traffic contract table (Figure 25). The values shown in the figure are the values used throughout the experiments. Once the attributes in this table have been set to the desired

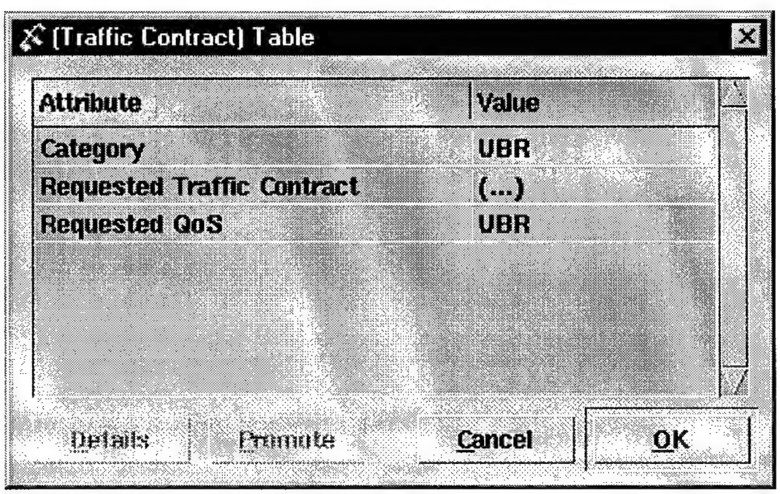


Figure 25. Traffic contract table

settings, the "requested traffic contract" attribute must be edited. When doing this, a window containing the requested traffic contract table will appear (Figure 26). This table allows for the customization of the PCR, MCR, SCR, and MBS attributes (only the PCR attribute was modified for the experiments conducted). Upon editing this attribute,

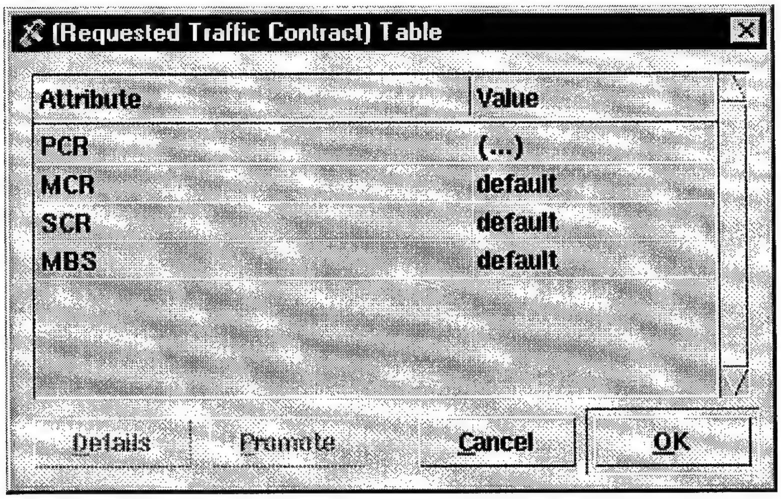


Figure 26. Requested traffic contract table

another window pops up (Figure 27) which provides access to the PCR attributes. Of these attributes, only the "incoming" attribute must be changed. In Figure 27, the value

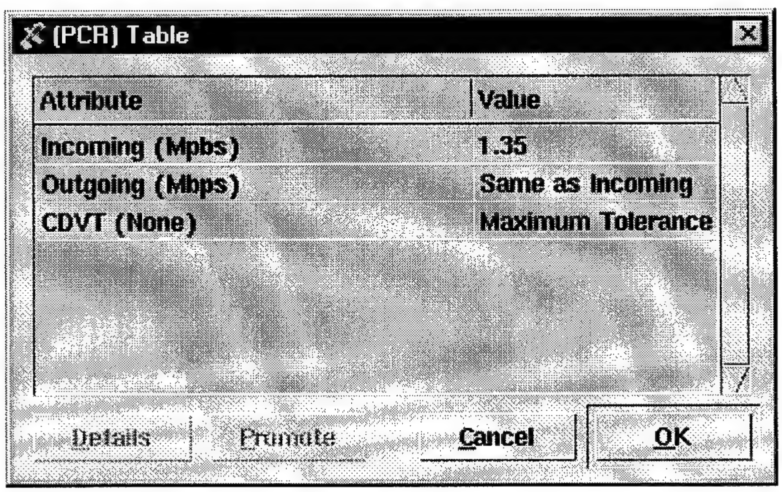


Figure 27. Peak cell rate table

"1.35" is used to specify a T1 data rate over the PVC.

Although a T1 link has a data rate of 1.544Mbps, the data rate actually achieved over the link is greater than the value specified for the "incoming" attribute and therefore a value must be selected that maximizes the data rate without exceeding the actual bandwidth of a T1 link. Trial and error demonstrated that "1.35" was the maximum

value that could be entered without exceeding the bandwidth capability of the T1 links in the network model. In the same manner, values of "7.2" and "40.0" were selected for 8 Mbps and T3 links respectively.

4.2.7 Simulation configuration

The simulation configuration provides standard options such as simulation duration, seed values, etc.; however, there is one particular option that is less than intuitive and is extremely critical in completing simulations in a timely fashion. When configuring a simulation, there is an attribute named "compound_cell_enabled." This attribute must be set to "enabled." If disabled, simulations will run much longer. For example, with "compound_cell_mode" disabled, one particular experiment that simulates two users downloading thirty 1MB files apiece took approximately 1.5 hours to complete. With "compound_cell_mode" enabled, this same simulation completed after approximately 8.5 minutes, less than a tenth of the previous time required. The "compound_cell_enabled" feature accomplishes this speedup by packaging multiple 53-byte ATM cells into a large virtual cell prior to transmission. This has the effect of reducing simulation overhead since fewer cells are transmitted.

4.3 Simulation Results

The remainder of this chapter presents the results obtained from the simulations. Section 4.3.1 discusses the validation process used to verify the correctness of the results returned by OPNET. Sections 4.3.2 – 4.3.4 present the analysis of the results for each scenario. Section 4.3.5 compares the performance of the three scenarios. Section 4.4 sums up the results and concludes this chapter.

4.3.1 Validation of OPNET results

Prior to conducting the experiments and gathering data, test simulations were conducted for the purpose of validating the results returned by OPNET. For these test simulations, a test model was built to simulate the baseline scenario with a single user requesting a download consisting of 30 image files. For the tests, request and response sizes were 10KB and 1MB respectively.

To determine if the application response time returned by the test simulations was as expected, an application response time was calculated analytically. Calculating an expected application response time required knowledge of the service times for each node and the transit times for data across each link in the network model. For example, if a server has an inter-request time of 1 second and 20 files are requested, the service time for that server is 19 seconds. For a link with a bandwidth of 1.544 Mbps, transmitting a 10KB file across the link (not counting propagation delay, transport protocol effects, etc.), takes 81920 bits/1,544,000bits/sec or approximately 50ms.

In order to determine which nodes and links in the model are utilized and how much data is crossing them, the scenario under simulation must be examined. This information is found in the manual configuration table. Using this table, the file sizes, nodes, and links being utilized are determined and application response time can be calculated by adding together the total server and transit times for the model (Table 4). Note that the calculated application response time assumes data transmitted by a source node is received at the destination node without any congestion or effects from TCP protocols. For this reason, the TCP "receive buffer size" (at each workstation and server node) was set to 36864 bytes to minimize the effects of TCP protocols during the test

simulations. This change made it easier to validate the results since calculating delays caused by TCP protocols is complex.

Given that the baseline scenario was designed for concurrent requests and responses to emulate real-world operations, it follows that the application response times from the test simulations should be less than the calculated application response time since the calculations also assume sequential execution. The test simulations confirmed expectations. The simulated application response times were less than the calculated application response time. Five test simulations were run and each simulation came back with the same value with a deviation of only nanoseconds. Table 6 shows the calculated application response time along with the mean response time returned by the test simulations.

For further confirmation, the model was modified to allow all requests and responses to occur in a serial fashion (emulating sequential execution). Now when the simulation is run, the

Table 6. Test "application response times"

Calculated "Application Response Time" (seconds)	OPNET "Application Response Time" (seconds)
209.6 seconds	186.2 seconds

resulting application response time should be close to the calculated application response time since the calculated application response time assumes sequential execution of tasks. This turns out to be the case. The application response time returned by the simulation is now approximately 208.3 seconds. Only slightly more than a second differentiates the two times. The discrepancy between the calculated time and returned simulation time is

probably because not all of the TCP effects were eliminated from the simulation. Further tests have shown that this can be accomplished by manipulating the TCP "receive buffer" size. As buffer size is increased, application response time decreased. The opposite is also true. This follows since the larger the buffer size the less congestion there will be on the network and thus fewer TCP interruptions. The buffer size can be manipulated to the correct value to eliminate TCP effects and obtain the calculated application response time. This was not done since the values obtained were sufficiently close to the calculated values to verify the correct behavior of the model. The results obtained in this validation process show OPNET does return expected results. Prior to conducting experiments, the TCP receive buffer was set back to OPNET's default value to allow TCP protocol effects to occur providing more realistic results.

4.3.2 Baseline scenario

Using the factors and associated levels presented in Chapter 3, simulation of the baseline scenario required 36 individual experiments. Each of these experiments represents a possible configuration of the baseline scenario. The results of each of the thirty-six experiments conducted are shown in Table 7. The raw data obtained from these experiments can be found in appendix B.

After collecting the application response times, an analysis was performed to determine if the results were statistically significant. For each experiment, a mean application response time and standard deviation was derived for the purpose of calculating a 90% confidence interval. Once confidence intervals were calculated a visual test was performed to determine if the results were statistically significant. The visual test is a performance evaluation method by which the confidence intervals of

different alternatives are plotted on a graph and the intervals are compared to see if they overlap. If the confidence intervals do not overlap, then one factor can be declared higher or lower than the other at the derived level of confidence. If the intervals do overlap and the mean of one is in the confidence interval of the other, then the alternatives are not different. Finally, if the confidence intervals overlap but no mean is in the confidence interval of the other, then further tests are required. Figure 28 shows a visual test comparing the results from experiments 1, 2, 7, and 8 (see Table 7). It is clear from the figure the results in columns one and two are different. That is, the confidence intervals (barely distinguishable) do not overlap. Although not discernable from the figure, the intervals for the values in columns three and four are extremely small and also do not overlap. Therefore, the application response times of these four experiments are different. This was the case for each of the baseline experiments. The visual tests

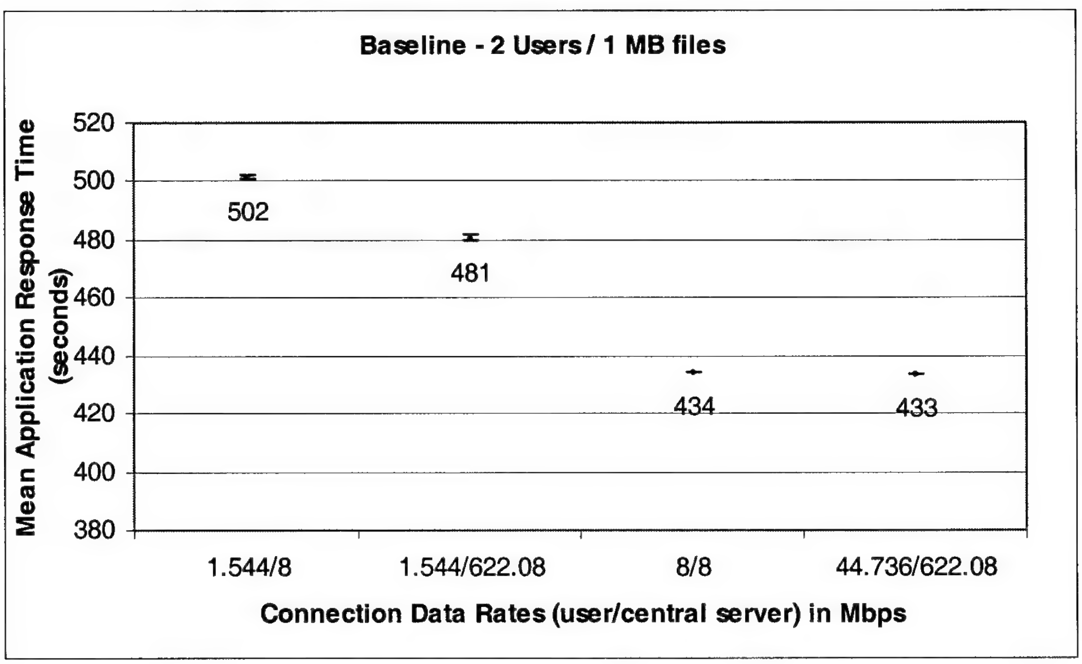


Figure 28. Visual test for experiments 1, 2, 7, and 8

Table 7. Baseline experiments

	Scenario	File Size	# Users	User Connection Bandwidth	Central Server Connection
1	1	1MB	2	1.544Mbps	8 Mbps
2	1	1MB	2	1.544Mbps	622.08Mbps
3	1	1MB	10	1.544Mbps	8 Mbps
4	1	1MB	10	1.544Mbps	622.08Mbps
5	1	1MB	20	1.544Mbps	8 Mbps
6	1	1MB	20	1.544Mbps	622.08Mbps
7	1	1MB	2	8 Mbps	8 Mbps
8	1	1MB	2	44.736Mbps	622.08Mbps
9	1	1MB	10	8 Mbps	8 Mbps
10	1	1MB	10	44.736Mbps	622.08Mbps
11	1	1MB	20	8 Mbps	8 Mbps
12	1	1MB	20	44.736Mbps	622.08Mbps
13	1	10MB	2	1.544Mbps	8 Mbps
14	1	10MB	2	1.544Mbps	622.08Mbps
15	1	10MB	10	1.544Mbps	8 Mbps
16	1	10MB	10	1.544Mbps	622.08Mbps
17	1	10MB	20	1.544Mbps	8 Mbps
18	1	10MB	20	1.544Mbps	622.08Mbps
19	1	10MB	2	8 Mbps	8 Mbps
20	1	10MB	2	44.736Mbps	622.08Mbps
21	1	10MB	10	8 Mbps	8 Mbps
22	1	10MB	10	44.736Mbps	622.08Mbps
23	1	10MB	20	8 Mbps	8 Mbps
24	1	10MB	20	44.736Mbps	622.08Mbps
25	1	100MB	2	1.544Mbps	8 Mbps
26	1	100MB	2	1.544Mbps	622.08Mbps
27	1	100MB	10	1.544Mbps	8 Mbps
28	1	100MB	10	1.544Mbps	622.08Mbps
29	1	100MB	20	1.544Mbps	8 Mbps
30	1	100MB	20	1.544Mbps	622.08Mbps
31	1	100MB	2	8 Mbps	8 Mbps
32	1	100MB	2	44.736Mbps	622.08Mbps
33	1	100MB	10	8 Mbps	8 Mbps
34	1	100MB	10	44.736Mbps	622.08Mbps
35	1	100MB	20	8 Mbps	8 Mbps

performed for the rest of the experiments can be found in Appendix A.

An analysis of variation (ANOVA) was conducted to determine what percentage of variation in application response times could be attributed to a given factor and if that factor was statistically significant (detailed analyses are located in Appendix C). Table 8 presents the results of the ANOVA for the baseline simulation. Based upon these results, the most obvious conclusion that can be drawn is that file size (accounting for over 91% of the variation) completely overwhelms any variations resulting from changes in user connection or central server connection bandwidths. The analysis shows with respect to file size, all other factors are negligible in their impact on application response time.

Table 8. Factor contribution towards application response time variations

Factor/Factor interactions	% of variation
file size	91.97%
number of users	.52%
user connection speed	.71%
central server connection speed	1.29%
file size/number of users	.64%
file size/user connection speed	1.04%
file size/C.S. connection speed	1.69%
number of users/user connection speed	.09%
number of users/C.S. connection speed	.52%
user connection speed/C.S. connection speed	.21%
Percent variation accounted for	*98.68%*
Percent variation not accounted for	*1.32%*

It was expected that increasing file size would cause application response time to increase, however it was not known in advance that this factor would so completely overwhelm the others. Figures 29 through 31 illustrate this. Figures 29 through 31 show the trend in application response times as the number of users and speeds of the user and

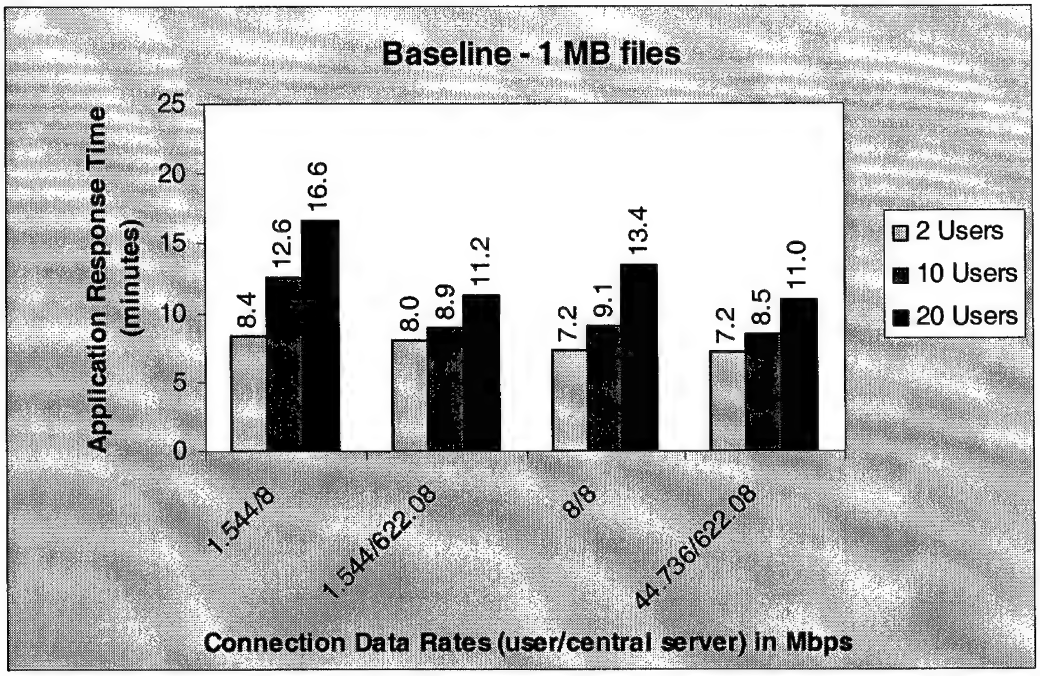


Figure 29. Application response time trends for 1MB files

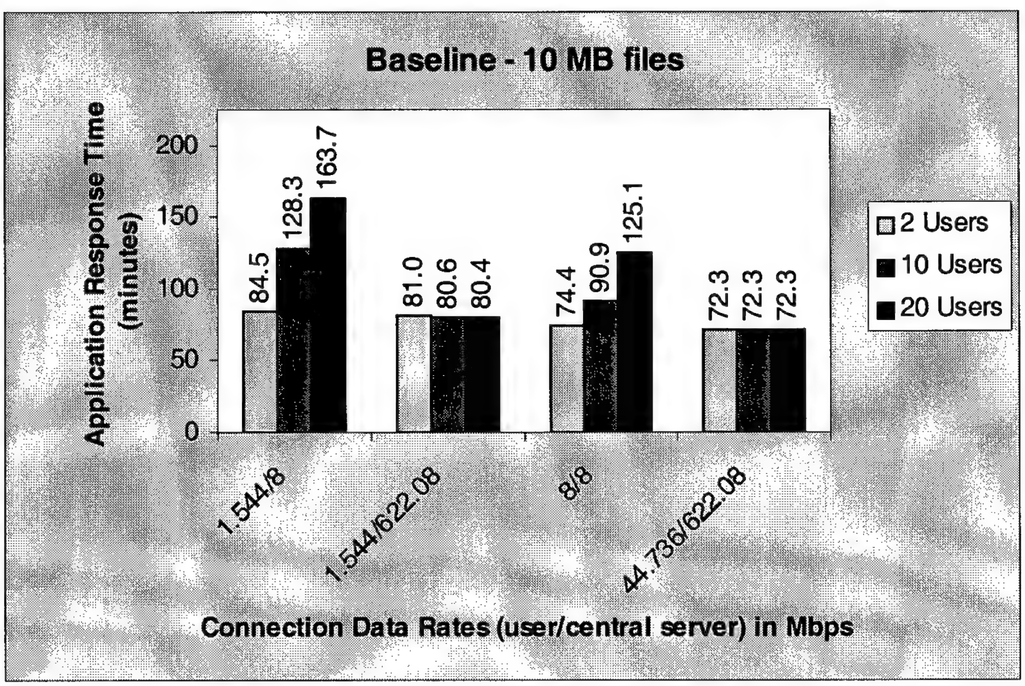


Figure 30. Application response time trends for 10MB files

central server connections vary while file size is held constant. These figures illustrate what the ANOVA test confirmed, that application response times were not greatly impacted by these factors. In fact, the speedup from the worst-case configuration to the best-case configuration is not even linear. In Figure 29, the worst-case scenario is the configuration where the user connection bandwidth is 1.544Mbps and the central server connection bandwidth is 8Mbps. The best case consists of a user connection bandwidth of 44.736Mbps and a central server connection bandwidth of 622.08Mbps. This is nearly a 29-fold increase in user bandwidth and a 77-fold increase in central server connection bandwidth. The corresponding increase in application response time is still negligible.

For a different view of the data, Figure 32 illustrates the variation in application response time resulting from changes in file size. As the file size was increased by an

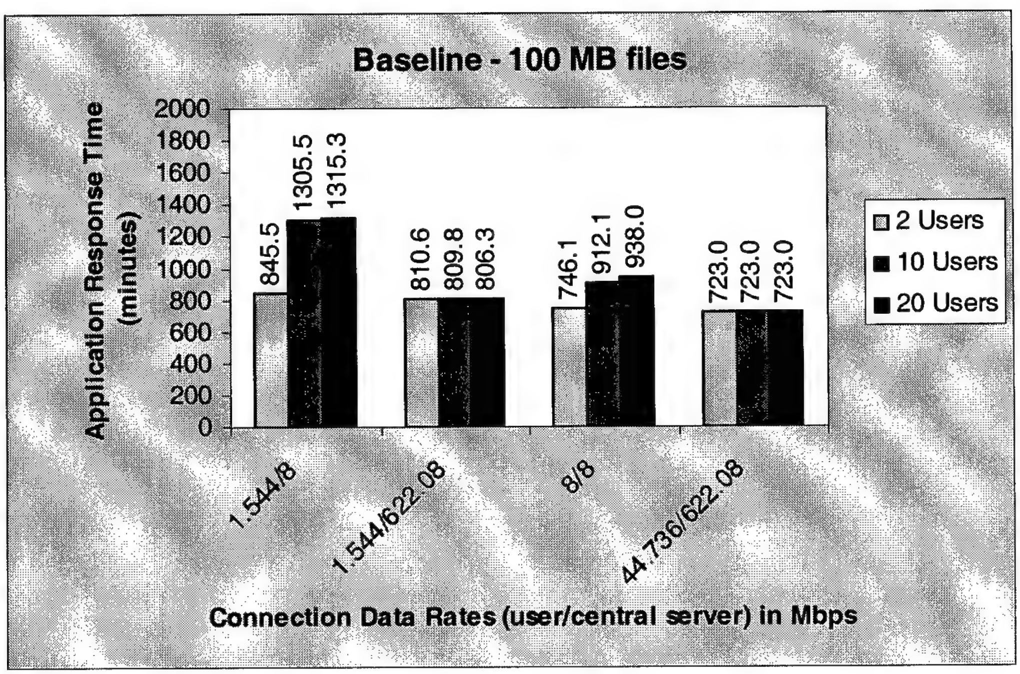


Figure 31. Application response time trends for 100MB files

order of magnitude, application response time showed a corresponding order of magnitude increase. This makes it easy to visualize how these order of magnitude

variations easily overwhelmed the modest variations caused by the other factors. These results highlight the need to narrow the analysis. Increases in file size and number of users is obviously going to increase the application response time, however, since these factors so completely overwhelm the other factors, a more focused analysis is required in order to determine the impact the different connection bandwidths have on application response time. For this reason, an ANOVA test was performed on the same data while factoring out file size and number of users. Performing the analysis in this manner provided insight into the variations caused by the user connection bandwidth and the

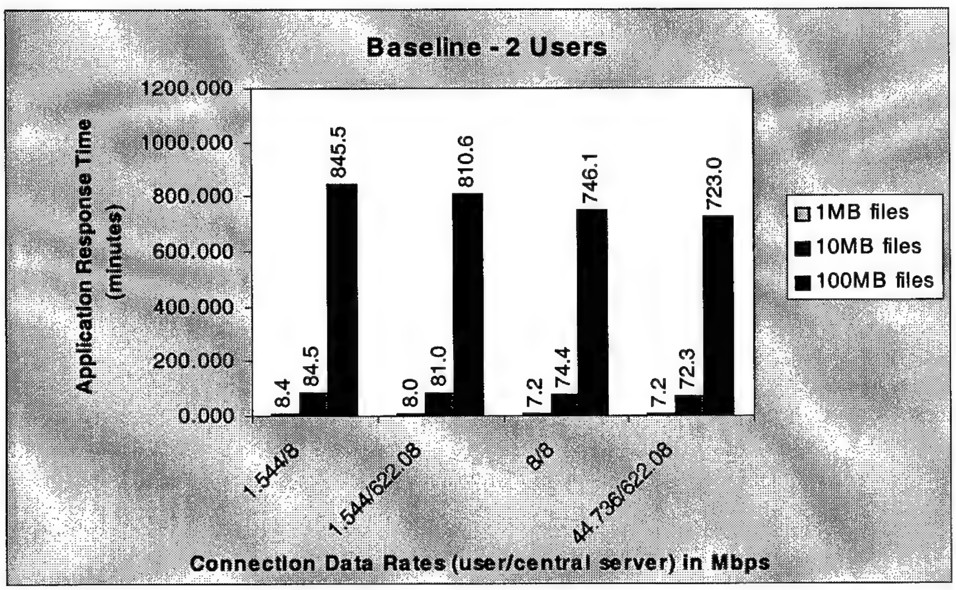


Figure 32. Application response time trends for varying file sizes

central server connection bandwidth. Upon re-accomplishing the analysis with file size and number of users factored out, the variations in application response time caused by user and central server connection bandwidths were more pronounced and more importantly, statistically accurate.

Table 9 shows the percentage of variation caused by the user connection bandwidth versus the central server connection bandwidth for each possible configuration

of file size and number of users. The results indicate that apart from the interaction of the two factors, the user connection bandwidth has a considerably greater impact on application response time than does the central server connection bandwidth. The variation explained by the user connection ranged from 33.4 – 41.6% whereas for the central server connection the range was only .03 to 14.3%. This follows since the bandwidth of the user connection always had the lowest bandwidth of any of the links in any of the experiments. The higher variations caused by the interaction of the two factors are due to links with higher bandwidths feeding links with lower bandwidths as is the case in each of the experiments. This increases the amount of buffering and TCP effects that occur.

Upon first glance, this information might not seem very useful. It should be intuitive the performance of any circuit in a network will be limited by the portion of the circuit with the lowest bandwidth. What these results do show, however is that despite an increase in the bandwidth of the central server connection from 8Mbps to 622.08Mbps (a 77-fold increase), the effects of this increase were very modest. The improvement in

Table 9. Percent variation caused by connection speed factors

configuration	% variation caused by user connection speed	% variation caused by central server connection speed	% variation caused by interaction of both factors
2 users/1MB files	34.7	.03	65.2
2 users/10MB files	34.4	.06	65.5
2 users/100MB files	34.4	.07	65.5
10 users/1MB files	38.7	2.2	58.9
10 users/10MB files	41.1	5.5	53.5
10 users/100MB files	41.6	5.7	52.7
20 users/1MB files	33.4	4.0	62.6
20 users/10MB files	35.8	14.3	49.9
20 users/100MB files	40.5	6.3	53.2

application response time as a result of this increase in bandwidth was all but negated due to the low bandwidth capability of the user connection. These results show that only modest improvements in application response time can be achieved by increasing the central server bandwidth from 8Mbps to 622.08Mbps when VDL users are utilizing connections with lower bandwidths (T1 – T3 range).

4.3.3 Scenario 2 (centralized storage and processing)

All experiments conducted for this scenario are shown in Table 10. In this scenario, file size is not a factor since the request and response sizes do not change. All requests are 1GB and all responses are 1MB. As a result, only twelve experiments were

Table 10. Scenario 2 experiments

Experiment #	Scenario	File Size	# Users	User Connection	Central Server Connection
37	2	1MB	2	1.544Mbps	8 Mbps
38	2	1MB	2	1.544Mbps	622.08Mbps
39	2	1MB	10	1.544Mbps	8 Mbps
40	2	1MB	10	1.544Mbps	622.08Mbps
41	2	1MB	20	1.544Mbps	8 Mbps
42	2	1MB	20	1.544Mbps	622.08Mbps
43	2	1MB	2	8 Mbps	8 Mbps
44	2	1MB	2	44.736Mbps	622.08Mbps
45	2	1MB	10	8 Mbps	8 Mbps
46	2	1MB	10	44.736Mbps	622.08Mbps
47	2	1MB	20	8 Mbps	8 Mbps
48	2	1MB	20	44.736Mbps	622.08Mbps

required as opposed to the thirty-six required for scenarios one and three. As with the baseline scenario, 90% confidence intervals were derived for the resulting data and an ANOVA was conducted. The derived confidence intervals indicated a slight difference in the results when compared to the results of the baseline simulation.

Figures 33 and 34 show that experiments where user connection and central server connection bandwidths were set at 8Mbps and 8Mbps respectively, visual tests

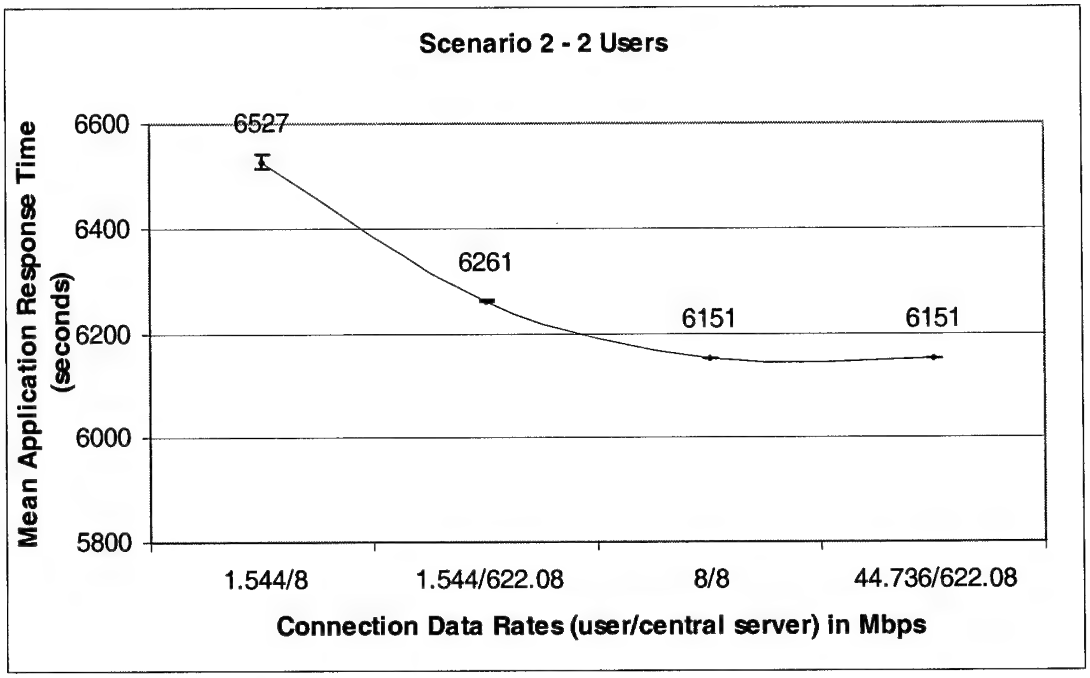


Figure 33. Visual test for 2-user configurations

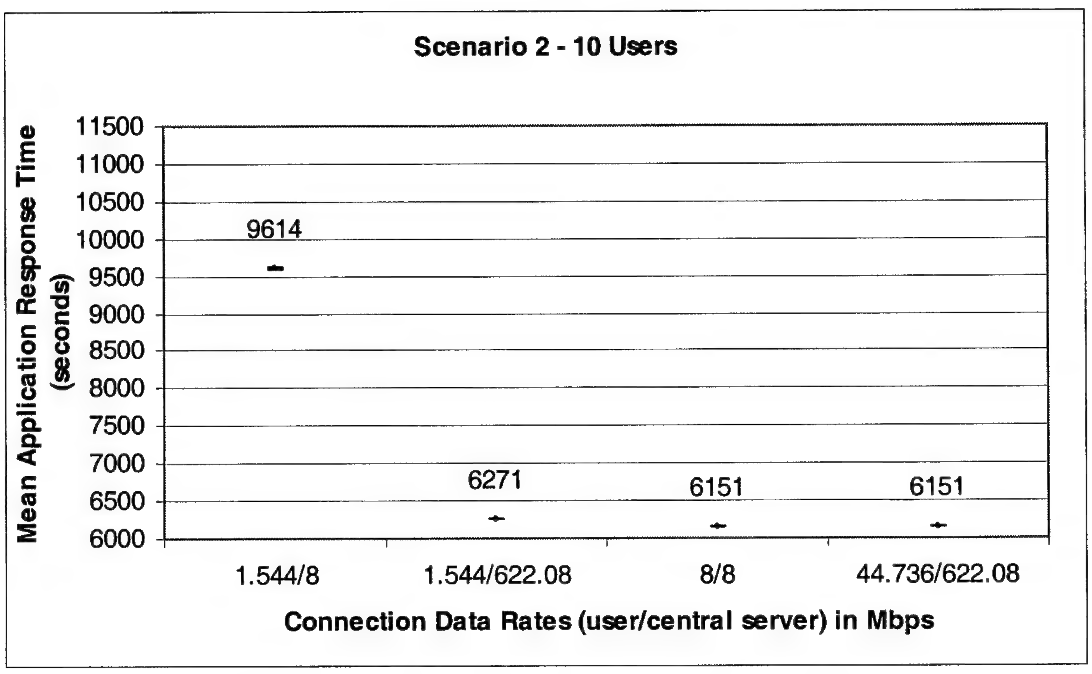


Figure 34. Visual test for 10-user configurations

confirmed the results were not statistically different from those where the same connection bandwidths were set to 44.736Mbps and 622.08Mbps respectively. Figure 35

shows the exceptions, those experiments involving 20 users. So, for the experiments consisting of less than twenty users and the configurations discussed above, nothing can be concluded regarding their performance with respect to one another. Statistically they

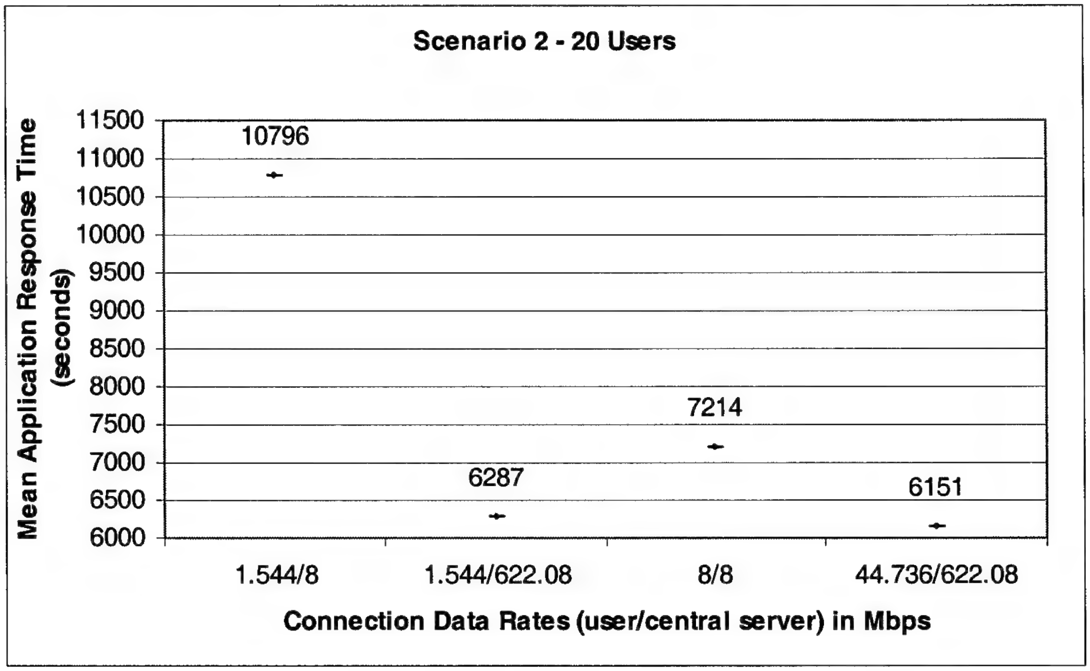


Figure 35. Visual test for 20-user configurations

are the same and one configuration cannot be said to be better or worse with respect to performance.

Performing an ANOVA on the results for 20 users yielded the percentages shown in Table 11. Like the baseline results, this analysis doesn't provide a clear picture of the impact of varying connection bandwidths. The percentages are somewhat skewed from a connection bandwidth standpoint since the majority of the files (20 out of 30) downloaded by the users come from the central server and the number of users is factored into the analysis. This makes it difficult to come to any accurate conclusions regarding the impact connection bandwidths have on application response time. In order to acquire

Table 11. Factor contribution towards application response time variations

Factor/Factor interactions	% of variation
number of users	13.5%
user connection speed	18.9%
central server connection speed	26.3%
number of users/user connection speed	6.2%
number of users/C.S. connection speed	13.3%
user connection speed/C.S. connection speed	15.6%
Percent variation accounted for	*93.8%*
Percent variation not accounted for	*6.2%*

a more accurate picture, the same process used in the analysis of the baseline results was applied – only the user connection and central server connection bandwidths were considered.

Based upon the percentages shown in Table 12, the same conclusion reached in the baseline analysis also applies to this scenario. The effects of significant increases in the central server connection bandwidth are nearly negated by the user connection bandwidth, which is limited to a maximum bandwidth of 44.736Mbps. An explanation

Table 12. Percent variation caused by connection speed factors

configuration	% variation caused by user connection speed	% variation caused by central server connection speed	% variation caused by interaction of both factors
2 users	**27.9**	**.02**	**72**
10 users	**42.1**	**2.5**	**55.4**
20 users	**39.7**	**5.7**	**54.6**

for the higher variations caused by interaction of the factors was provided in the previous section and applies to these results as well. Again, these results indicate only modest or negligible performance benefits can be achieved by increasing the bandwidth of the central server connection. Figure 36 supports this conclusion. Presented in this manner,

the bar graph demonstrates that despite significant increases in connection bandwidths, application response time improvements were very modest or non-existent.

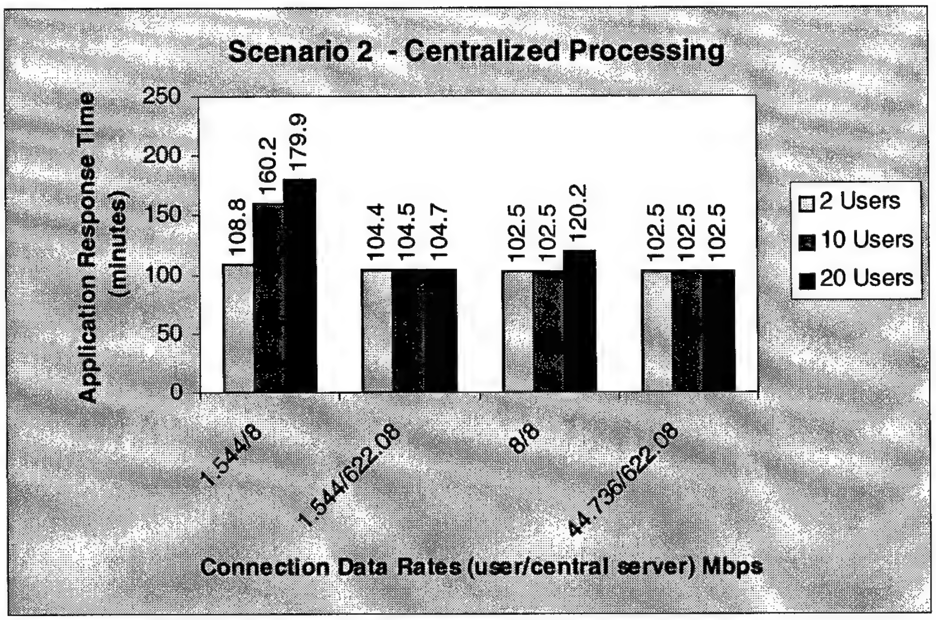


Figure 36. Application response time comparison for all configurations

4.3.4 Scenario 3 (direct download)

Scenario 3 consisted of the same experimental configurations shown in Table 7. Simulations of this scenario produced results possessing the same characteristics and nearly the same values as those produced in scenario 1. An ANOVA conducted on the results of these two scenarios (Appendix C, Figure C25) showed that variations caused by the scenario factor were statistically insignificant. Therefore, it cannot be said that either of these scenarios performed better or worse with respect to application response time. Thus, the same conclusions were drawn for this scenario regarding performance. For this reason, no further discussion or analysis of this scenario is required. The data for this

scenario can be found in Appendix B, visual tests are in Appendix A, and the associated ANOVA charts are in Appendix C.

4.3.5 Comparison of scenarios

With analyses of each individual scenario completed, an analysis of how the scenarios performed with respect to one another was conducted. The purpose of this analysis was to determine if one scenario provided better application response times than the other two scenarios. In accomplishing this analysis, only three factors were considered: traffic pattern scenario, bandwidth of the user connection, and bandwidth of the central server connection. File size and number of users were not considered for reasons discussed in previous analyses.

An ANOVA shows the traffic pattern scenario factor accounts for approximately 53% of the variation (see Table 13). This variation can be explained by examining the

Table 13. Variation percentages per factor

Factor	Percentage
Scenario	53%
User Connection B.W.	5.9%
Central Server Connection B.W.	.005%
Scenario/User Connection B.W.	7.3%
Scenario/Central Server Connection B.W.	.006%
User Connection B.W./Central Server B.W.	14.59%

mean application response time for each scenario (see Figure 37). This figure shows scenario two had a much smaller mean application response time than scenarios one and three. This difference accounts for the high percentage of variation shown in Table 13. The lower mean application response time is due to the fact that file size is not a factor in scenario two. For example, when twenty users (the maximum in the experiments) are

concurrently sending and receiving files, no more than 20.02 gigabytes of data (in scenario two each user transaction involves 1.001 gigabytes of data) will be traversing the

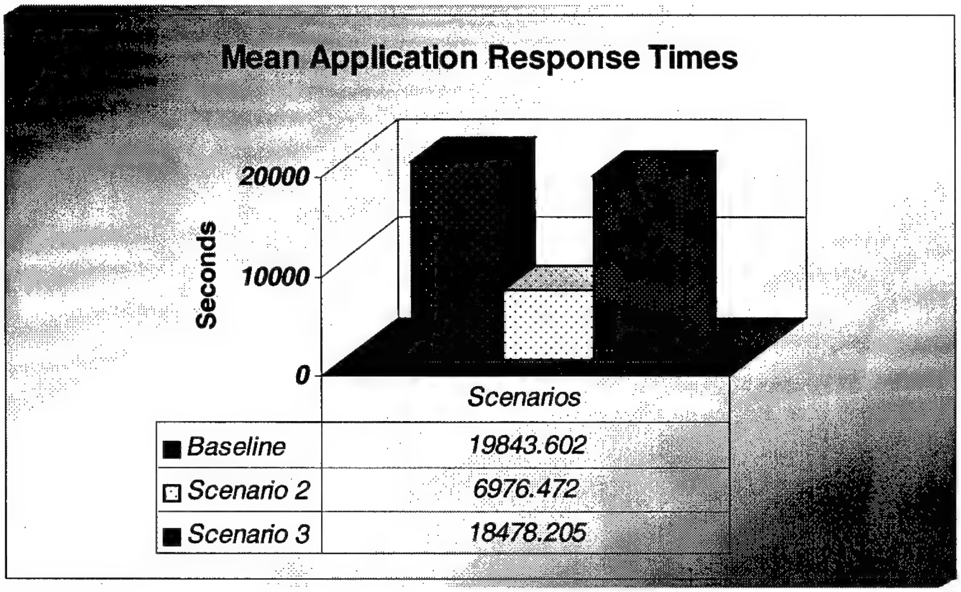


Figure 37. Mean application response time per scenario

network at any given time. In scenarios one and three, if twenty users are concurrently using the system and requesting the maximum of thirty 100MB files, 60 gigabytes of data could be traversing the network at any given time. This 60 gigabytes of data places a larger load on the system resulting in longer application response times. Data presented in the previous three sections confirms this. Although this performance trend is intuitive, it can nevertheless be used to determine which scenario performed the "best", which is not as obvious as it appears.

Determining which scenario performed best depends chiefly on the primary concern of VDL designers. That is, what is more important, decreasing the amount of traffic on the network or maximizing response time to the user? If response time is the primary concern, then the amount of data required by the user must be taken into

consideration. Figure 38 illustrates the maximum amount of data that may potentially be on the network at any given time for a given scenario, number of users, and file size.

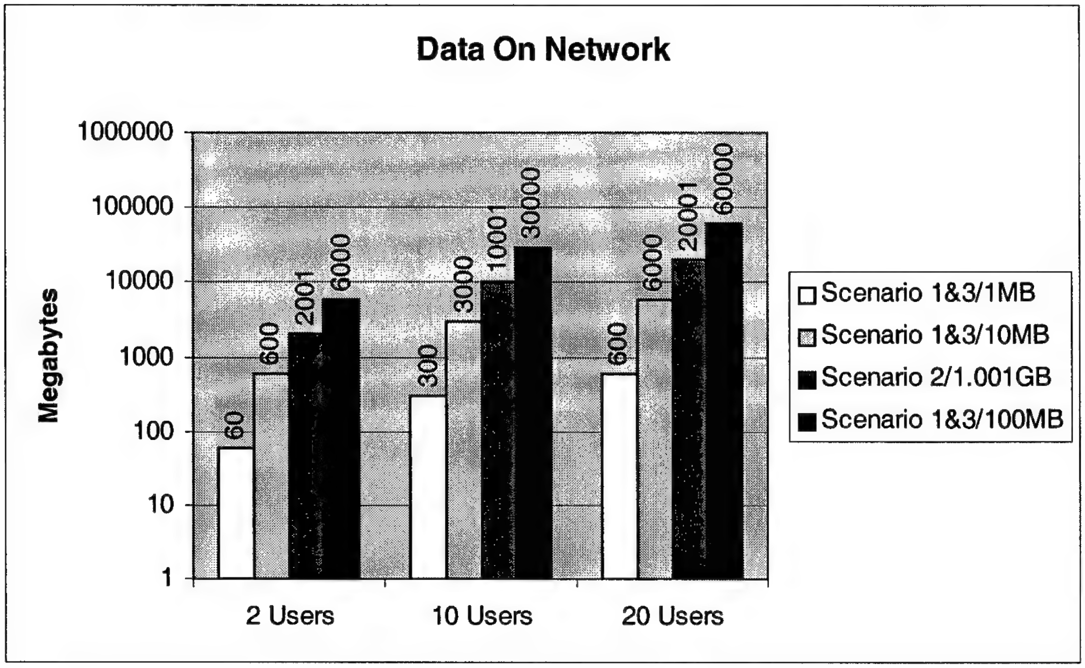


Figure 38. Data transfer amounts per scenario

Figure 39 shows the mean application response times correlating to the data amounts shown in Figure 38.

The comparison of the two figures illustrates that when scenarios one and three responded faster than scenario two, users were transmitting and receiving less data overall than the users in scenario two. This too is an intuitive performance trend, but it does show that the best scenario to select from a purely application response time perspective depends on how much data users will require on average to test their algorithms.

When selecting a scenario, if application response time is the primary consideration, then scenario one or three are preferred if the total amount of data required for downloading is less than the size of the algorithm file and the result file combined.

Otherwise, scenario two provides the best response to the user (not accounting for processing time at the MSRC).

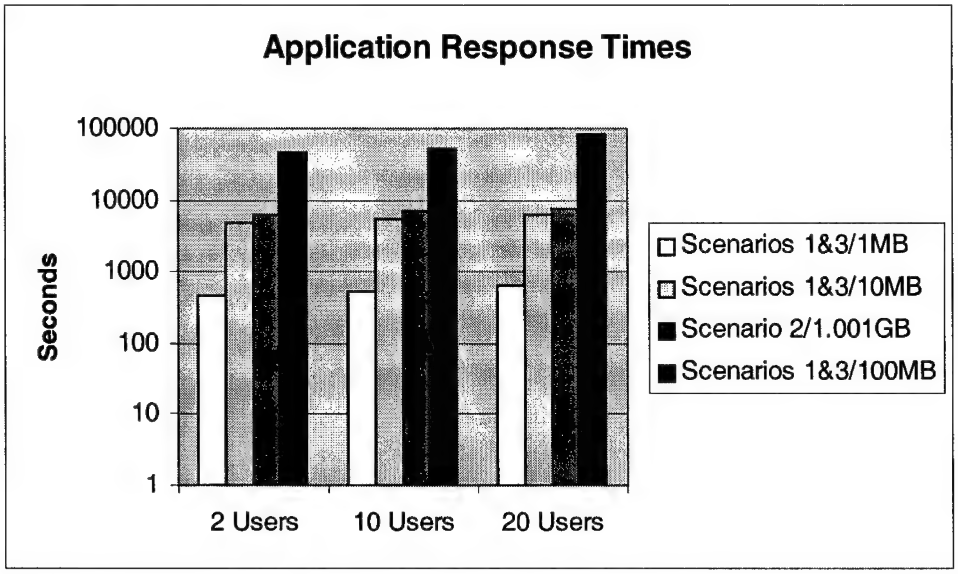


Figure 39. Application response times correlating to data in Figure 38.

Finally, if the primary goal is to minimize the number of files on the network, scenario two is the obvious choice since only algorithm files and result files are sent and received (two files per user). Scenario two also offers an advantage in response time if the amount of data users require for testing is on average greater than the size of the algorithm file and result file. Figures 38 and 39 illustrate this point. However, scenario two does have some major disadvantages. One obvious disadvantage is the high cost of retransmission. If the algorithm file does not arrive at the destination or is corrupted, then testing cannot occur and re-transmitting such a large file is extremely inefficient. In scenarios one and three, loss of a single or even a few files might not be as catastrophic to the user since testing can still proceed without the lost file(s). In other words, testing can proceed with the images that were received while the lost or corrupted files are re-sent.

4.4 Summary of Results

Applying the methodology presented in Chapter 3, three potential VDL collaboration scenarios were modeled and simulated. The results of these simulations showed expected performance trends and no statistical surprises were encountered. However, the results did lead to two conclusions. First, a significant increase in central server connection bandwidth results in very modest or negligible improvements in application response time. This demonstrates that unless VDL users possess similar bandwidth capabilities, improvements in application response time will be modest or negligible at best. The final conclusion comes from the comparison of the performance of the three scenarios. Scenario two provides the best response time to the user if the total amount of image file and signature data required for algorithm processing exceeds the total size in bytes of the algorithm and result files combined. If this is not the case, then either scenario one or three provides better response time.

5. Conclusions

5.1 Conclusions

With hundreds of automatic target recognition (ATR) researchers throughout the DoD participating in the Virtual Distributed Laboratory (VDL), a system is under development which will connect these researchers via a high-speed network called the defense research and engineering network (DREN). This network will allow the researchers to retrieve imagery and signature data located in data repositories dispersed throughout the DoD. Even more importantly, these researchers will be able to pool their effort and collaborate more easily and efficiently to develop better ATR algorithms for current and future combat systems. Additionally, it is anticipated this will save the DoD money through the reduction of redundant efforts. Despite the importance of this project, relatively little research has been conducted to determine the best way to configure the network for optimal performance. To help ensure success of the VDL, three potential collaboration scenarios were developed for the purpose of simulating anticipated workloads and system configurations. This was submitted as a method for providing designers of the VDL with performance trend data showing the impact certain design decisions have on simulated system performance (response time).

5.1.1. Analysis of Individual Scenarios

The three collaboration scenarios simulated were the baseline, direct-download, and centralized processing. Initial ANOVA tests performed on the results showed variances in application response time caused by file size completely overwhelmed variances caused by other factors. As a result, no conclusions could be reached regarding the impact of the other factors (specifically connection bandwidths) on application

response time. For this reason, further ANOVA analyses were conducted that focused strictly on two factors of primary interest -- user connection bandwidth and central server connection bandwidth. These analyses showed that in all three scenarios, the variance in application response time caused by changes in the user connection bandwidth dominated. Variances caused by changes in the central server connection bandwidth were negligible in comparison since the limiting factor was the low bandwidth of the user connection. While a thorough performance-cost analysis is required, these results indicate increasing the central server connection bandwidth from 8Mbps to 622.08Mbps will result in only modest or negligible performance gains if VDL users are limited to the lower bandwidths (1.544 – 44.736Mbps range).

5.1.2 Comparison of Scenarios

Finally, the three scenarios were compared to determine which delivered the best performance. Based upon ANOVA analyses and mean application response times, it was determined there was no difference between scenarios one and three with regards to application response time. The variance in application response time for these two scenarios was statistically insignificant. Scenario two however had a much lower application response time. Scenario two clearly performs better when the total size in bytes of the image files downloaded by the users in scenarios one and three exceeds the size in bytes of the algorithm files and results files combined. Conversely, scenarios one and three perform better if the total size in bytes of the downloaded image files does not exceed the total size in bytes of the algorithm and results files. These results and observations show that choosing the best scenario depends on the mean size of the data

sets required by users. Using the mean data set size and the observations above, a scenario can be selected that will maximize network response time.

5.2 Future Research

Simulations conducted for this research effort did not consider all performance aspects of the system under test. Some performance characteristics had to be estimated or derived from current knowledge since VDL specifications do not exist from which precise models could be built. For this reason, the primary concern of this research was to provide designers of the VDL with performance trend data for the purpose of aiding in the design decision process. Future research efforts might involve updating the models from this research with more precise information. For example, the server nodes used in the current models can be modified to account for database access times and actual processing times as measured on operational servers. Additionally, the central server and remote servers used in the models can be updated to reflect the actual hardware and software implementations once that information becomes available. Also, once all remote server locations are known, propagation delay can be built into the simulations as well. Two primary benefits would come from these enhancements to the current models. First, more accurate metrics will be attainable. Second, with precise models in place, any projected or contemplated changes in the system can easily be simulated to determine the impact on performance prior to implementing any changes to the system. Both of these benefits may ultimately save time and money when evaluating the impact of hardware, software, or configuration changes on system performance.

One final area of research worth examining is to investigate additional user and central server connection bandwidth configurations to determine those that provide

significant improvements in application response time. Currently, research has shown that significant increases in central server bandwidth result in modest or negligible improvements in performance when users only possess a 1.544Mbps – 44.736Mbps connection. This investigation would not be difficult using the models developed for this research.

5.3 Summary

In this research effort, a methodology was described for modeling and simulating three potential VDL network configurations. The performance trend data resulting from these simulations pointed out to VDL designers some potential performance issues that must be addressed as well as some future areas for research. As the VDL grows and evolves, more precise models of the VDL can be designed using the methodology applied in this research.

Appendix A: Visual Tests

The figures in this appendix contain the visual tests performed on the application response times obtained from the simulations. The interpretation of these visual tests is presented in Chapter 4, section 4.3.

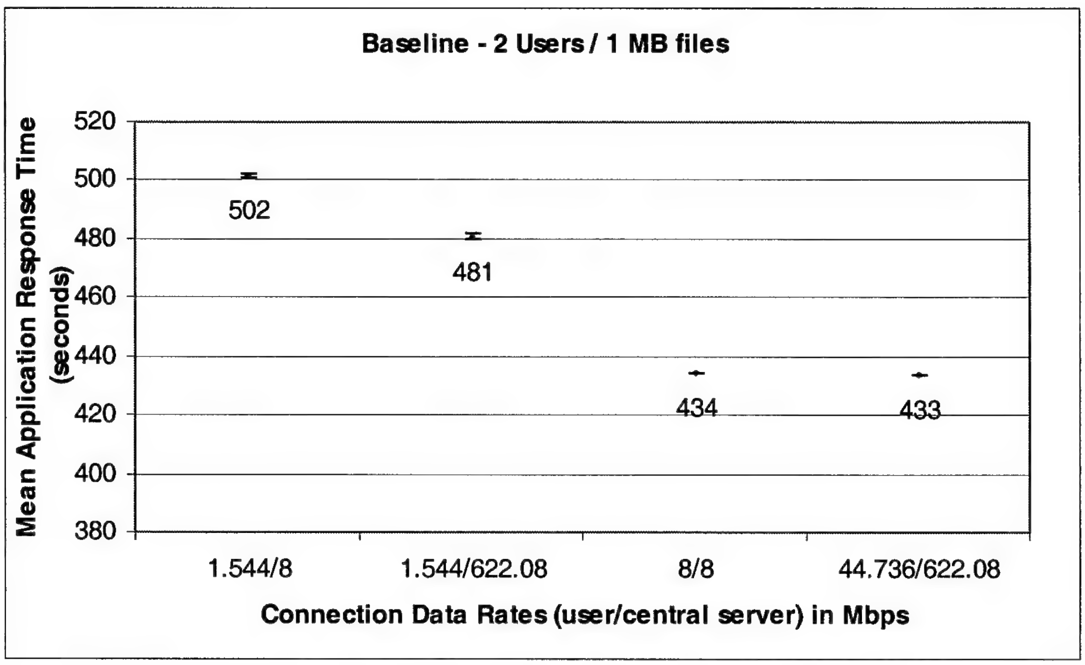


Figure A1. Baseline visual test for 2 users/1MB files

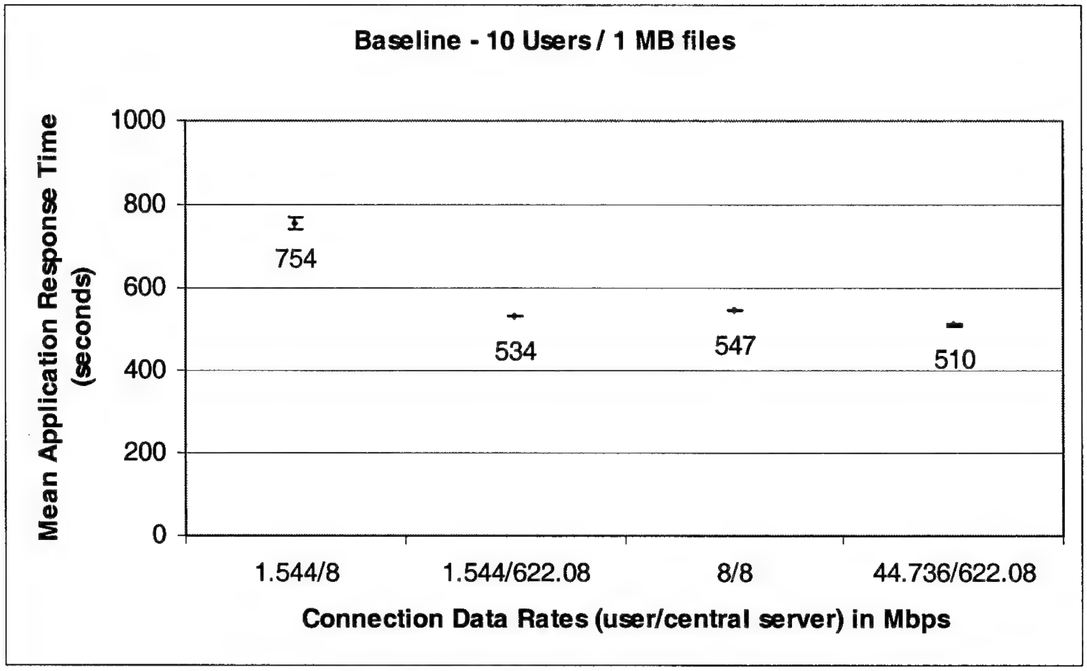


Figure A2. Baseline visual test for 10 users/1MB files

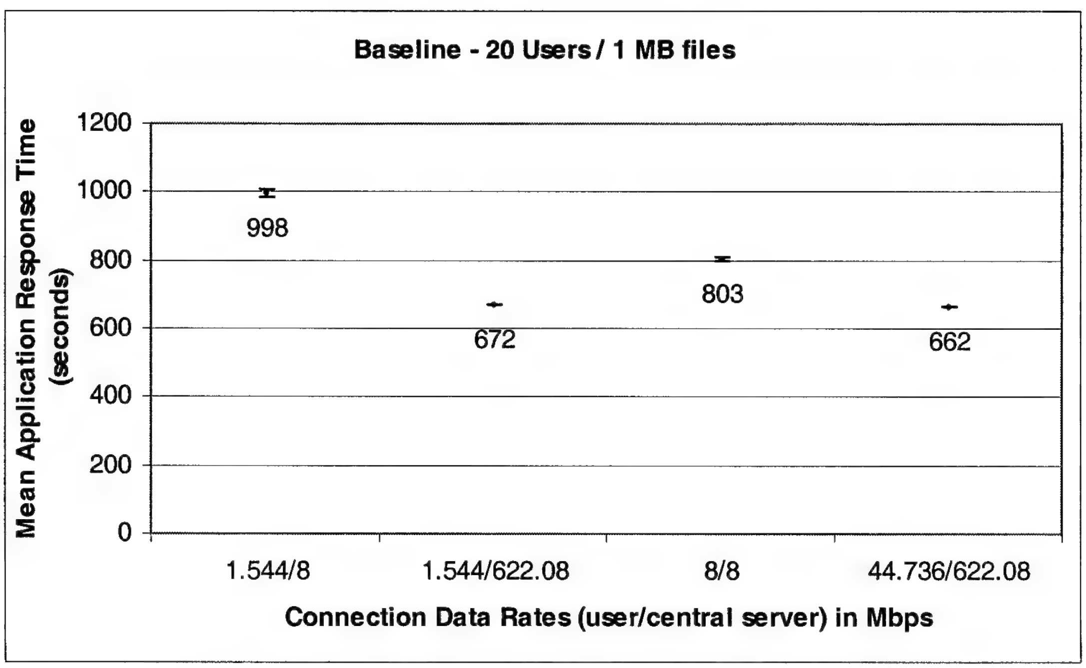


Figure A3. Baseline visual test for 20 users/1MB files

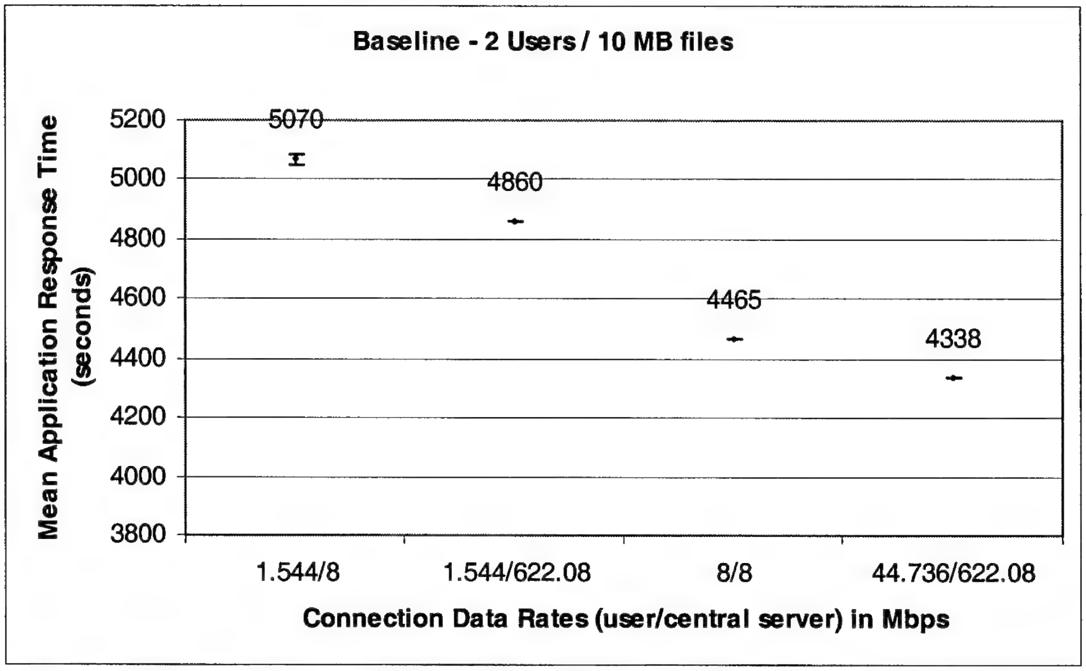


Figure A4. Baseline visual test for 2 users/10MB files

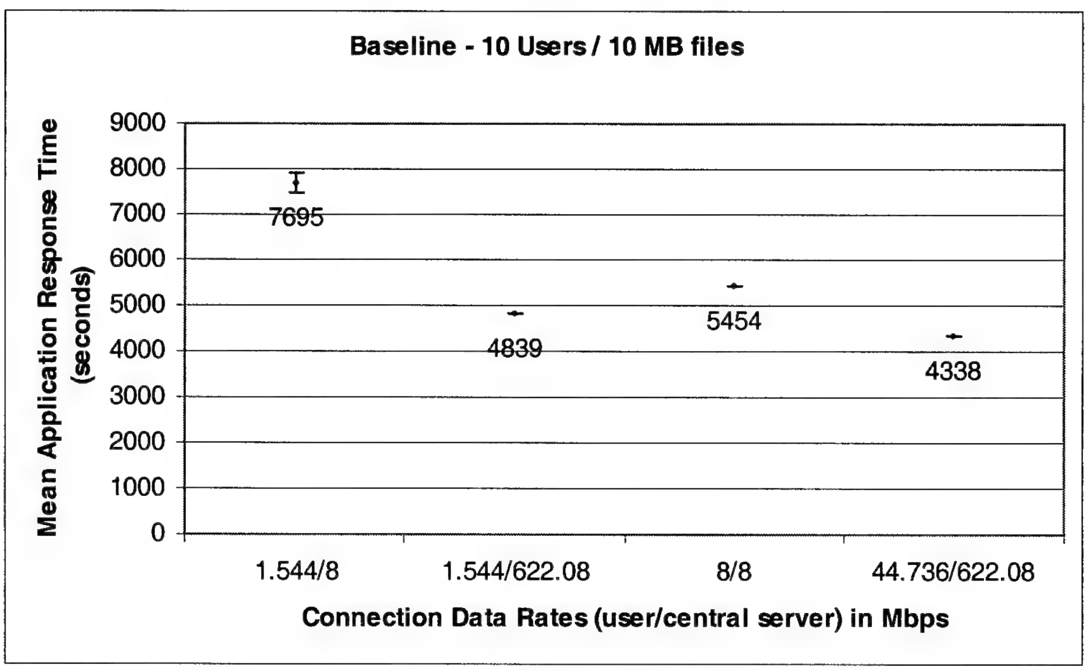


Figure A5. Baseline visual test for 10 users/10MB files

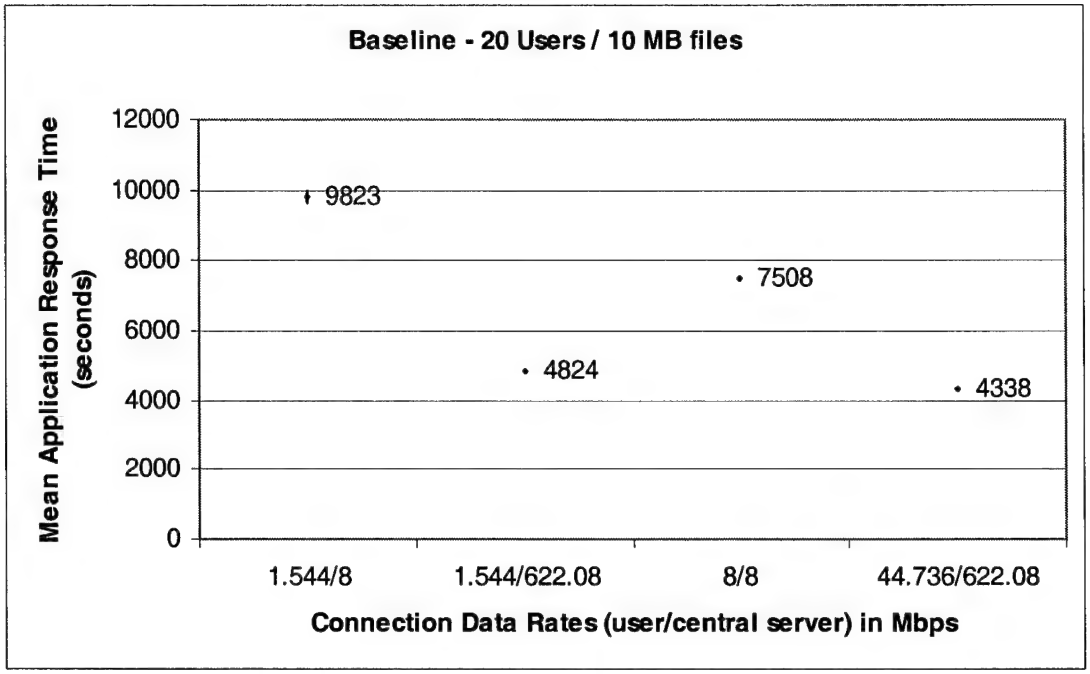


Figure A6. Baseline visual test for 20 users/10MB files

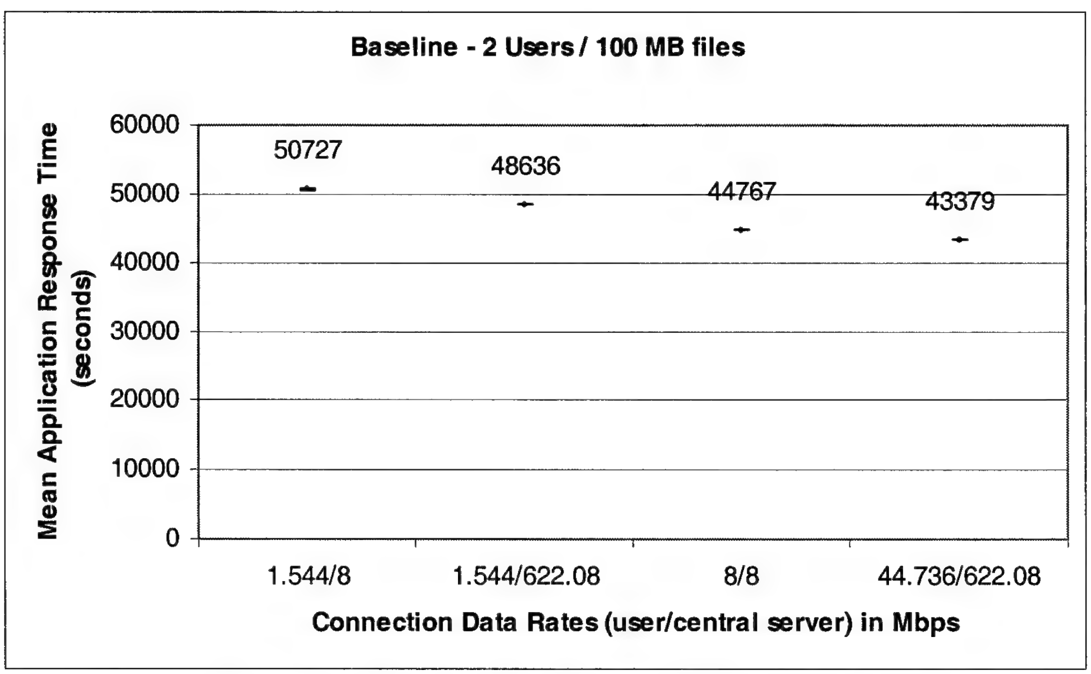


Figure A7. Baseline visual test for 2 users/100MB files

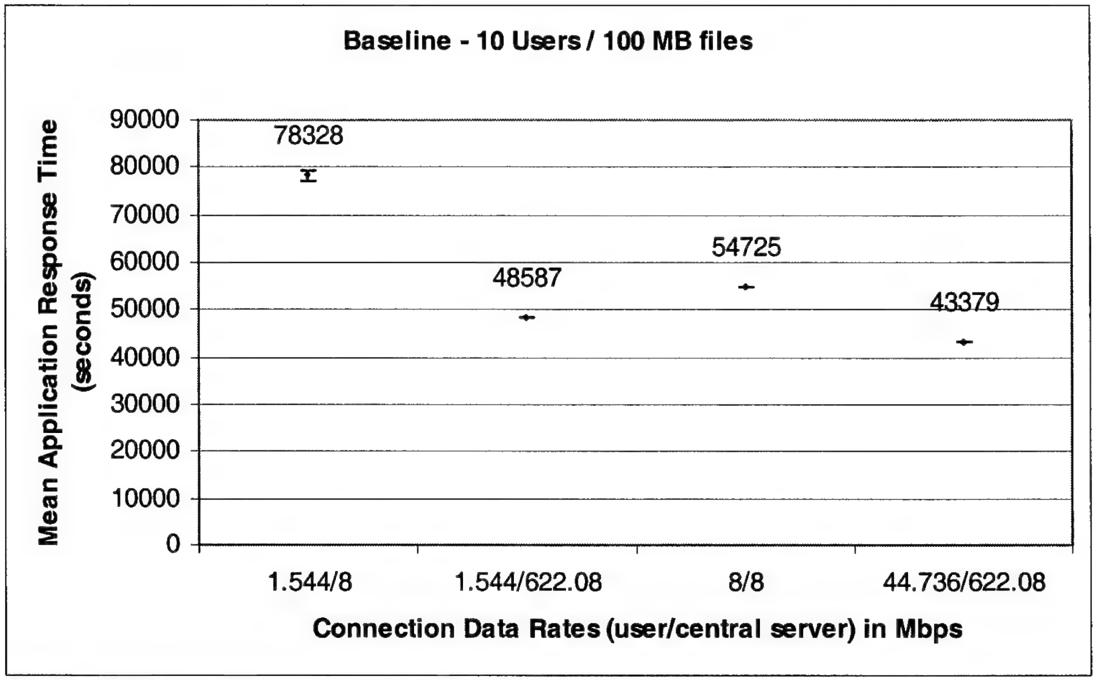


Figure A8. Baseline visual test for 10 users/100MB files

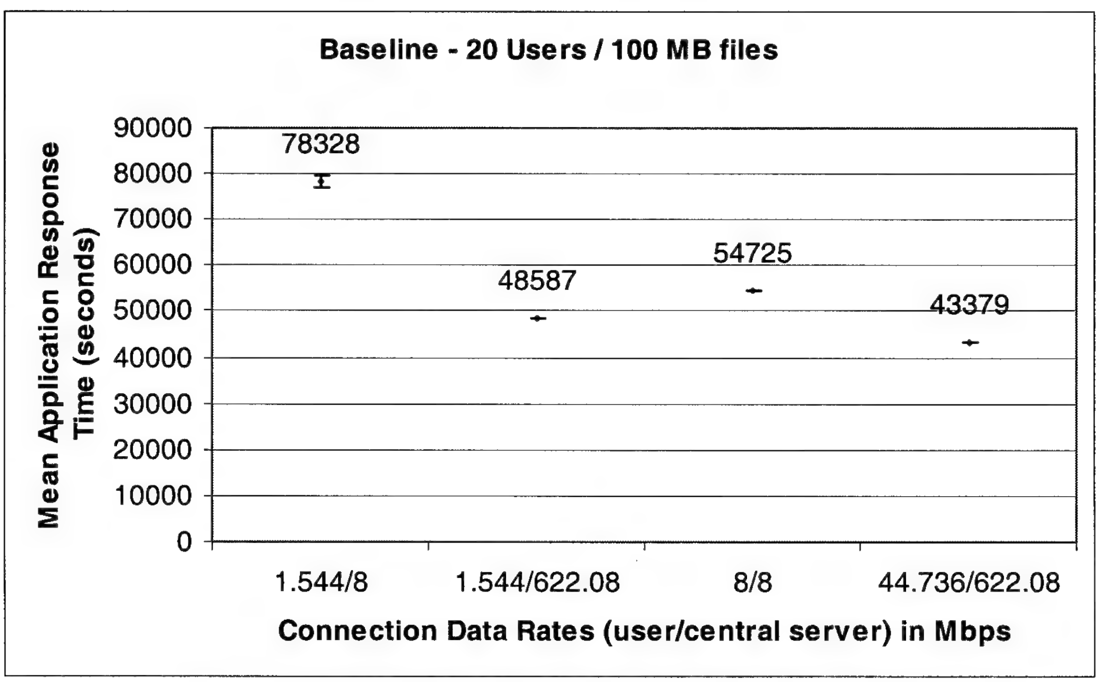


Figure A9. Baseline visual test for 20 users/100MB files

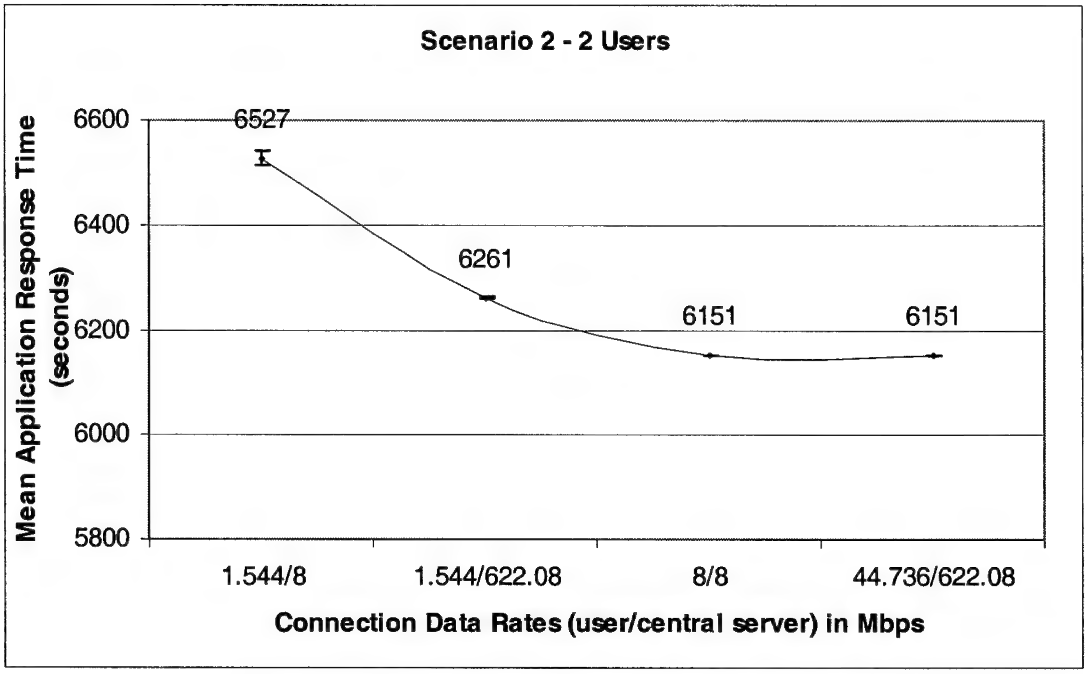


Figure A10. Scenario 2 – visual test for 2 users

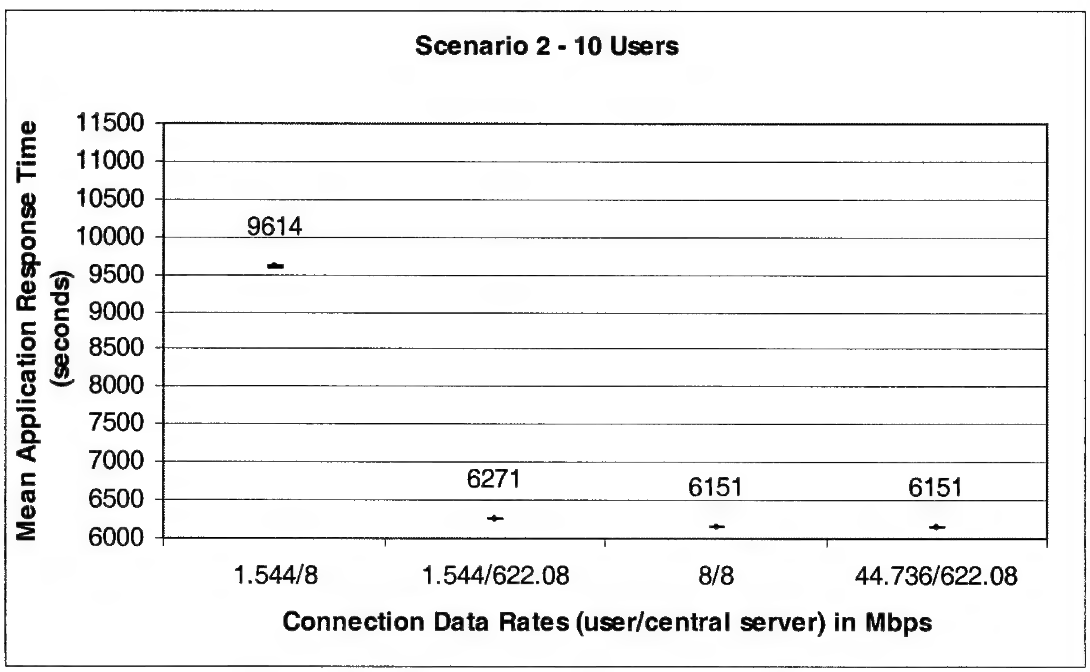


Figure A11. Scenario 2 – visual test for 10 users

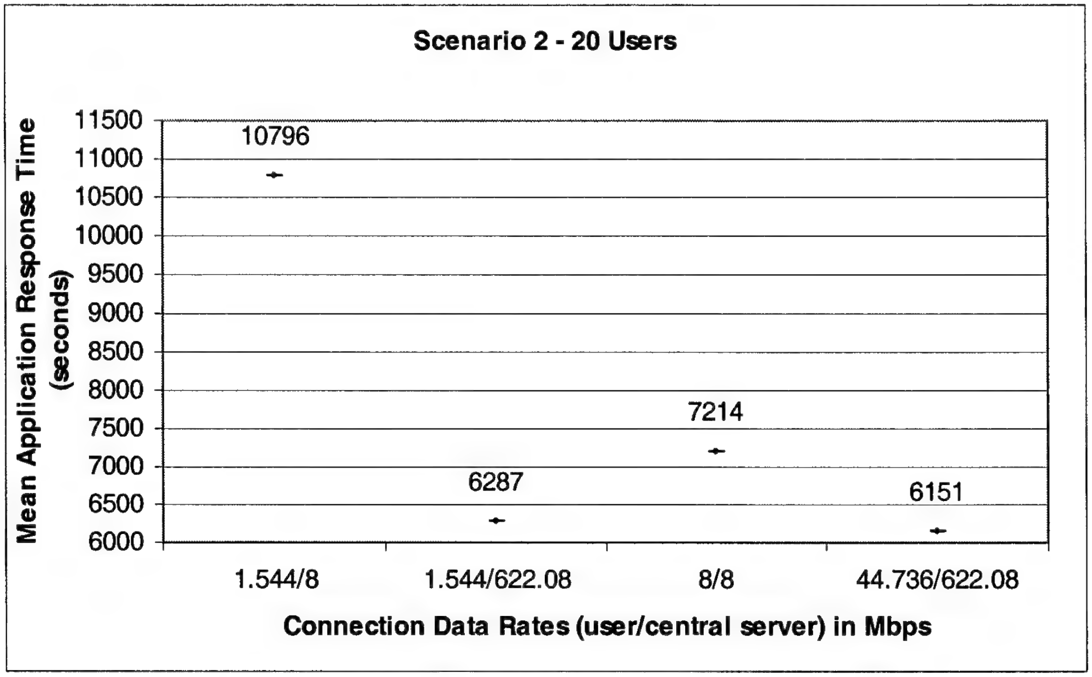


Figure A12. Scenario 2 – visual test for 20 users

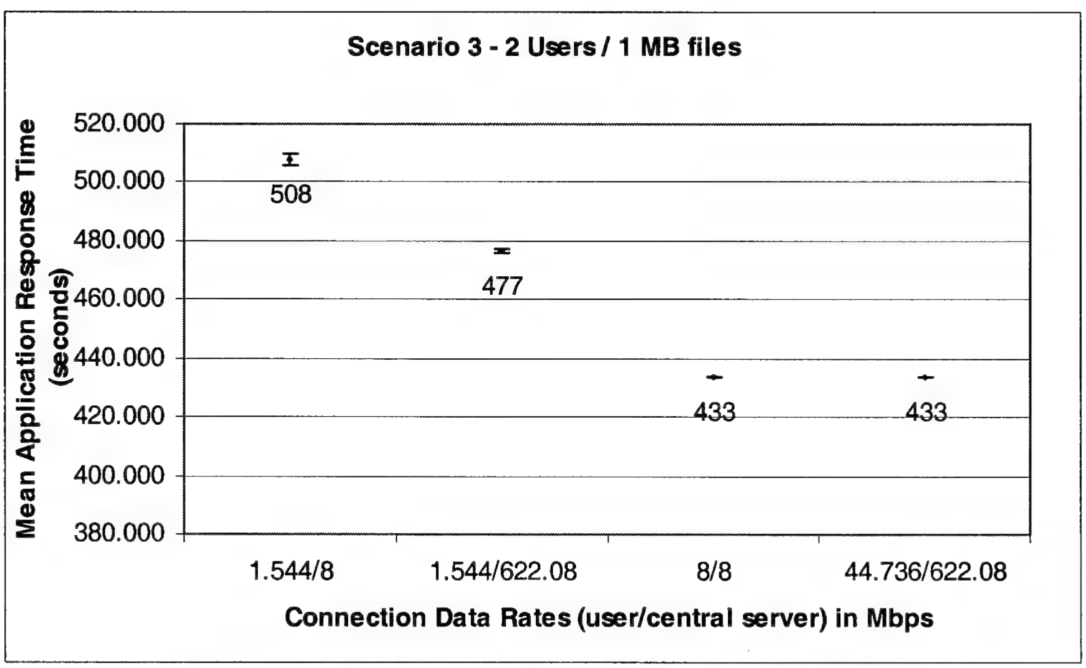


Figure A13. Scenario 3 - visual test for 2 users/1MB files

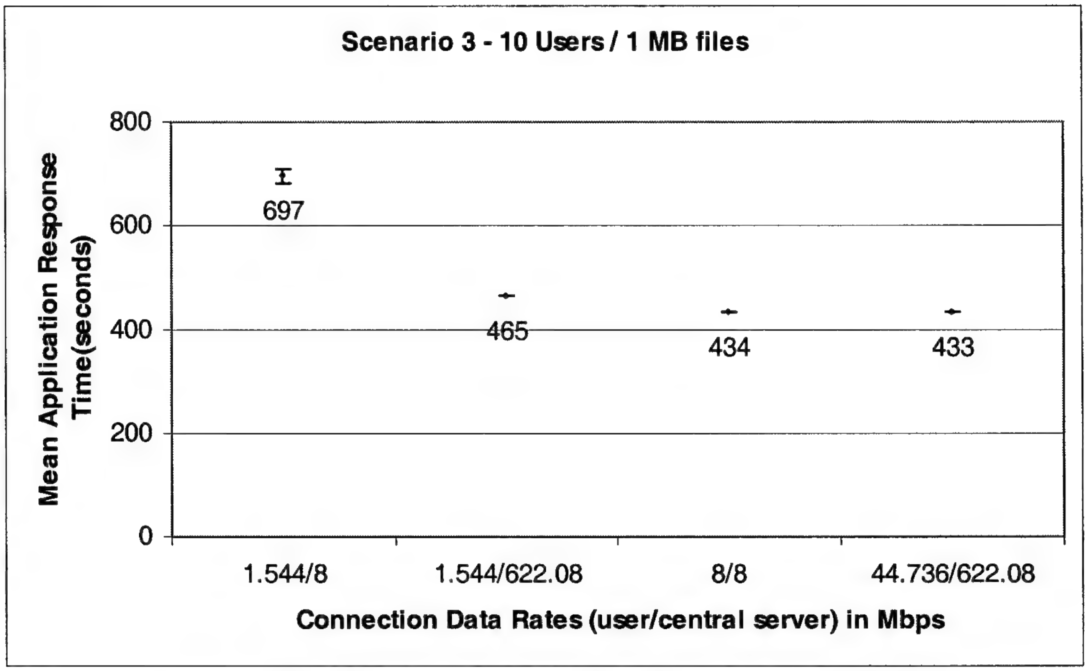


Figure A14. Scenario 3 - visual test for 10 users/1MB files

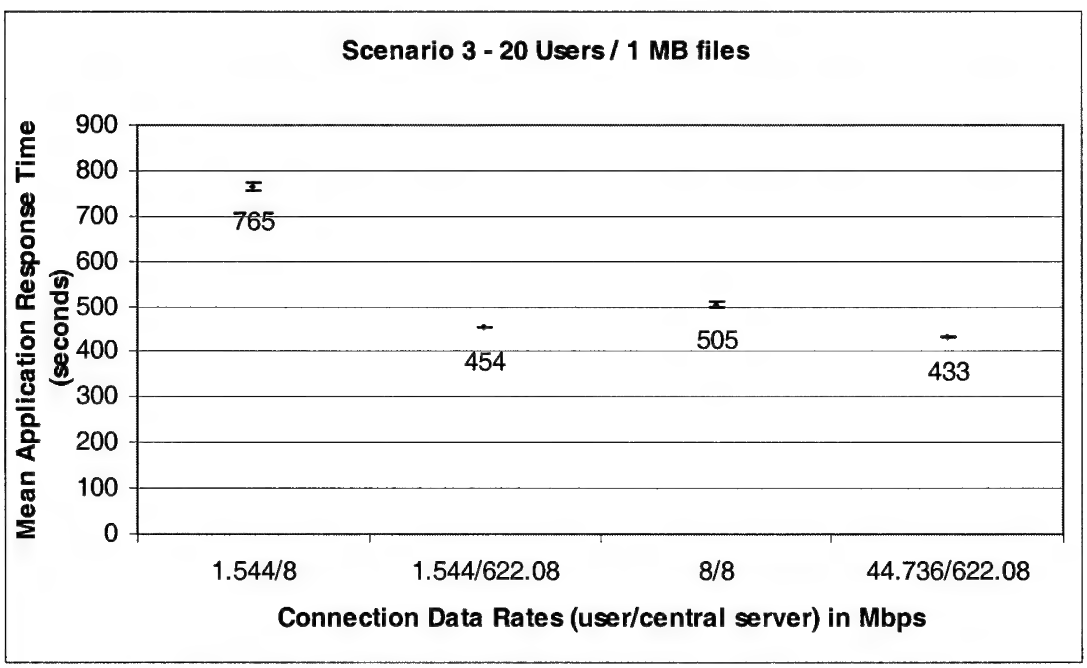


Figure A15. Scenario 3 - visual test for 20 users/1MB files

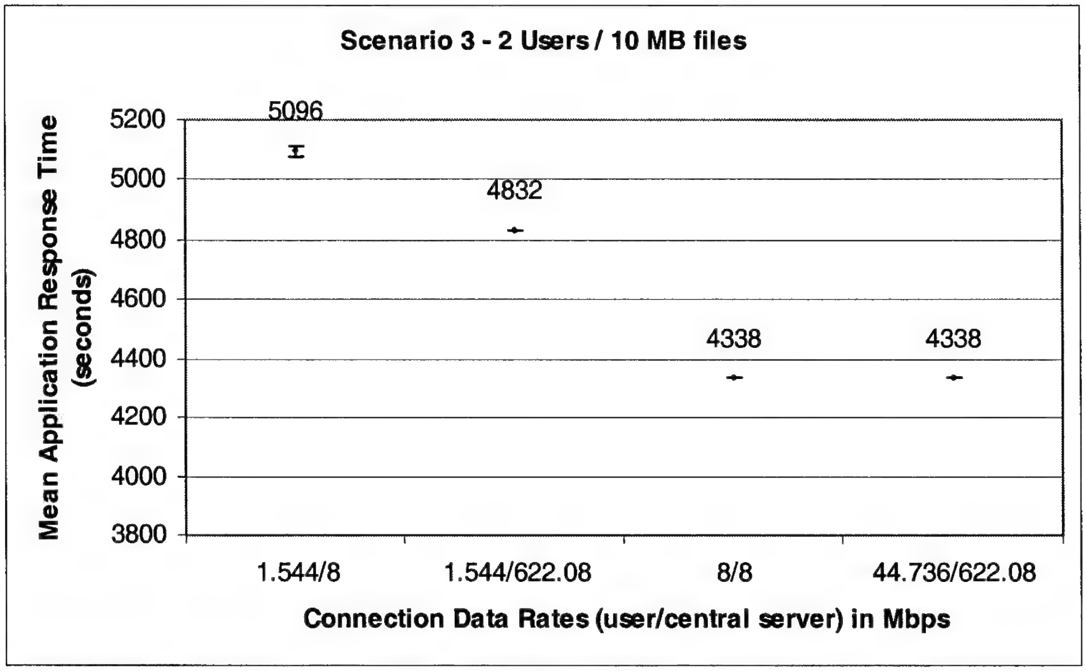


Figure A16. Scenario 3 - visual test for 2 users/10MB files

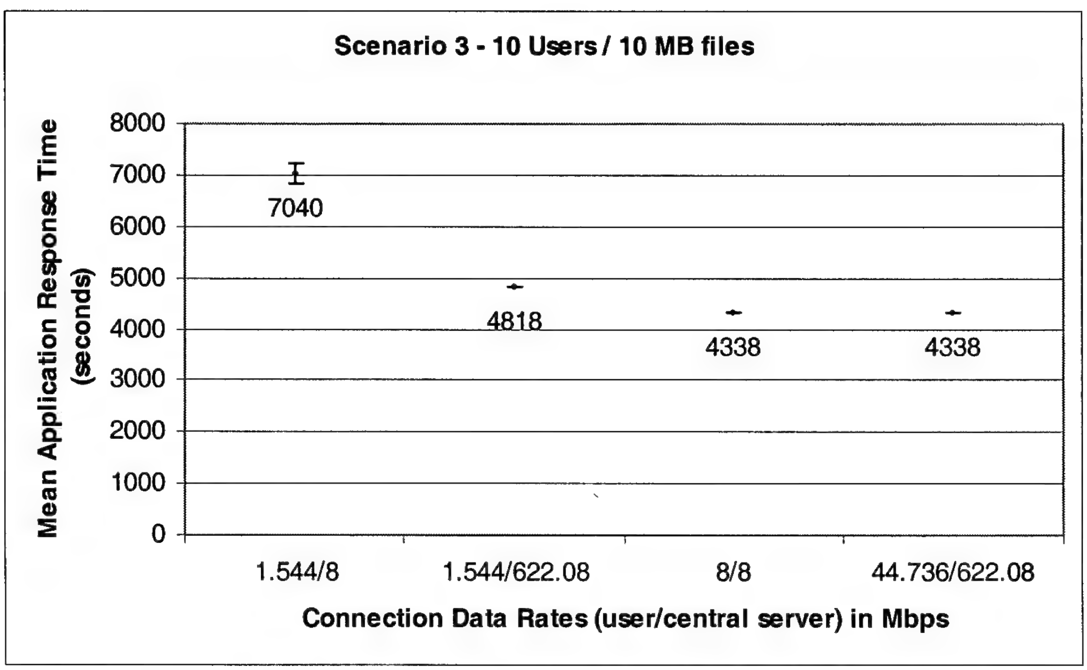


Figure A17. Scenario 3 - visual test for 10 users/10MB files

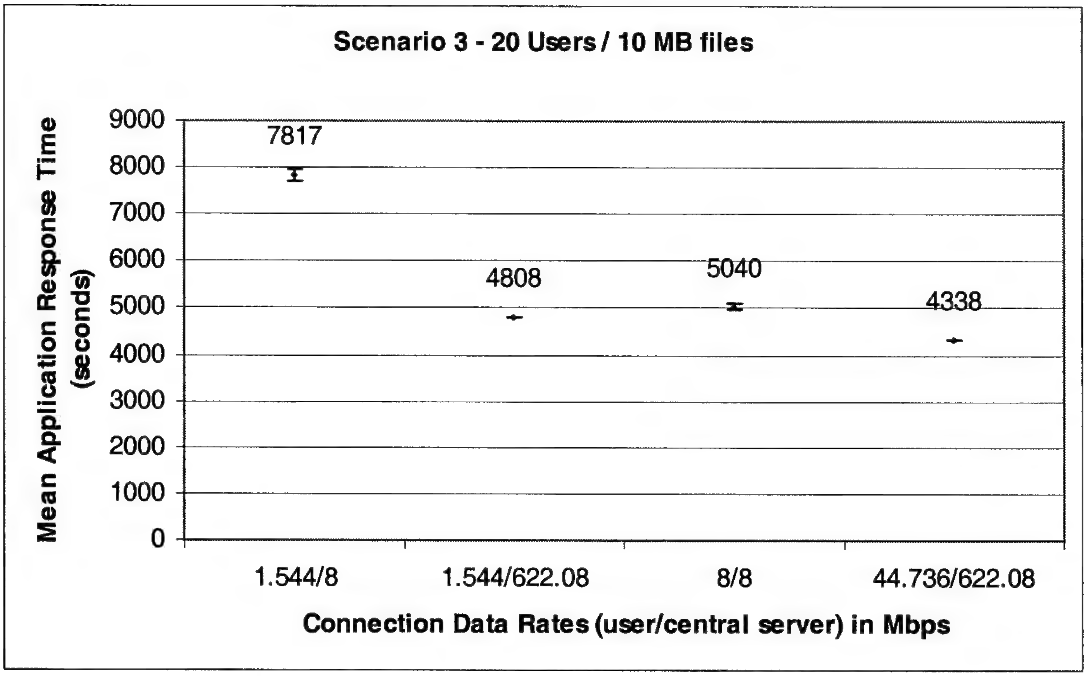


Figure A18. Scenario 3 - visual test for 20 users/10MB files

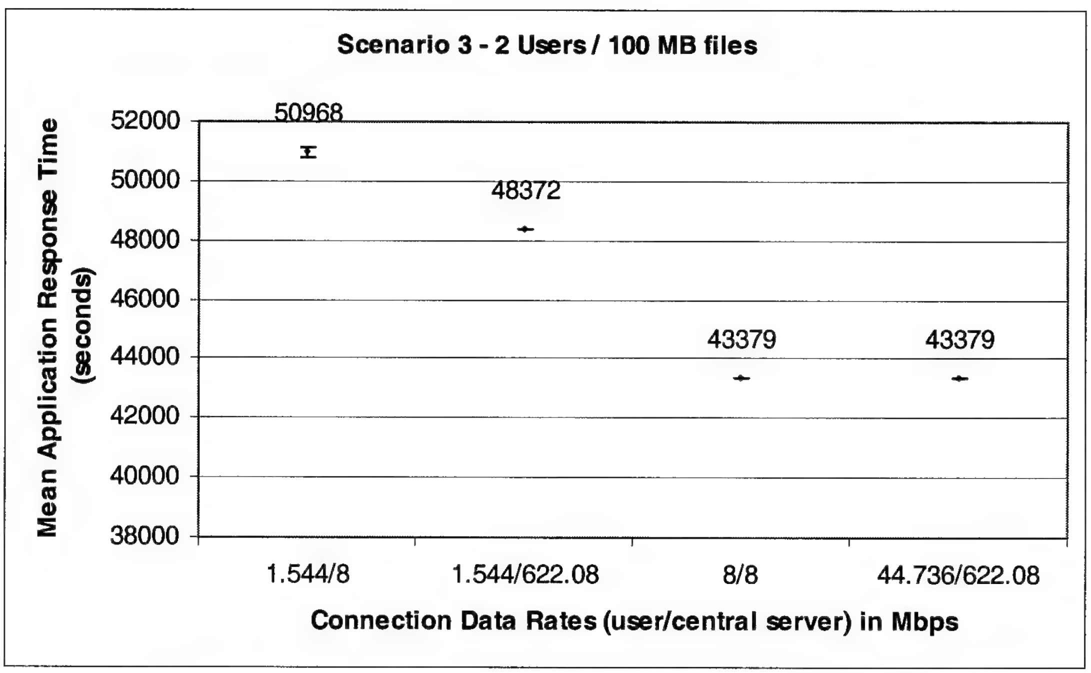


Figure A19. Scenario 3 - visual test for 2 users/100MB files

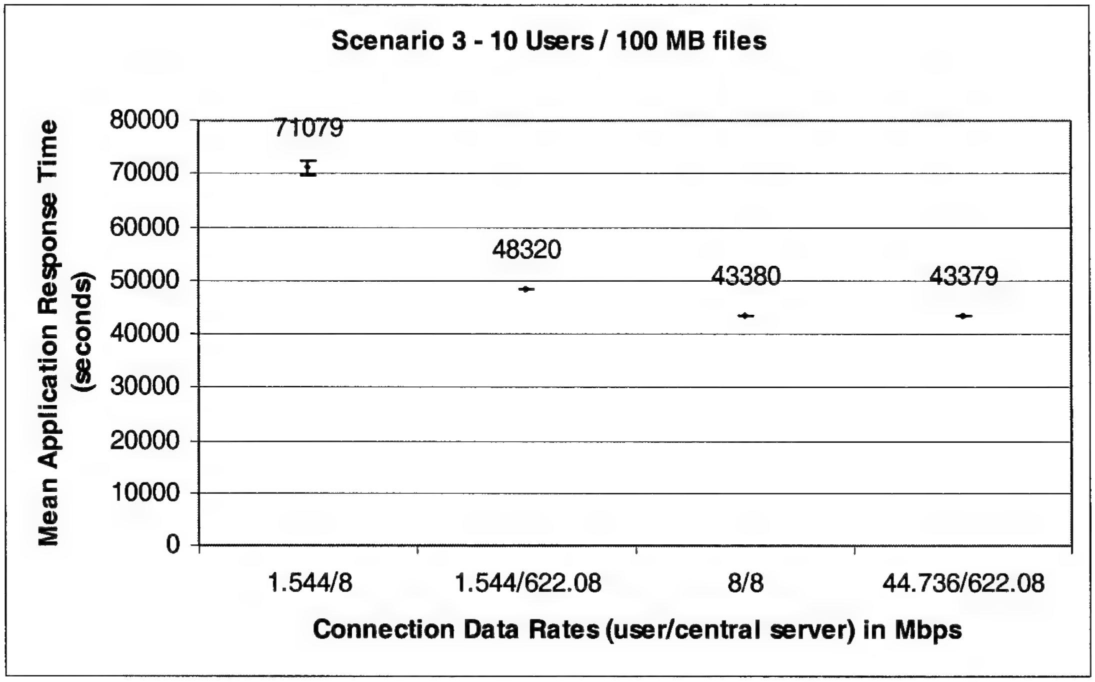


Figure A20. Scenario 3 - visual test for 10 users/100MB files

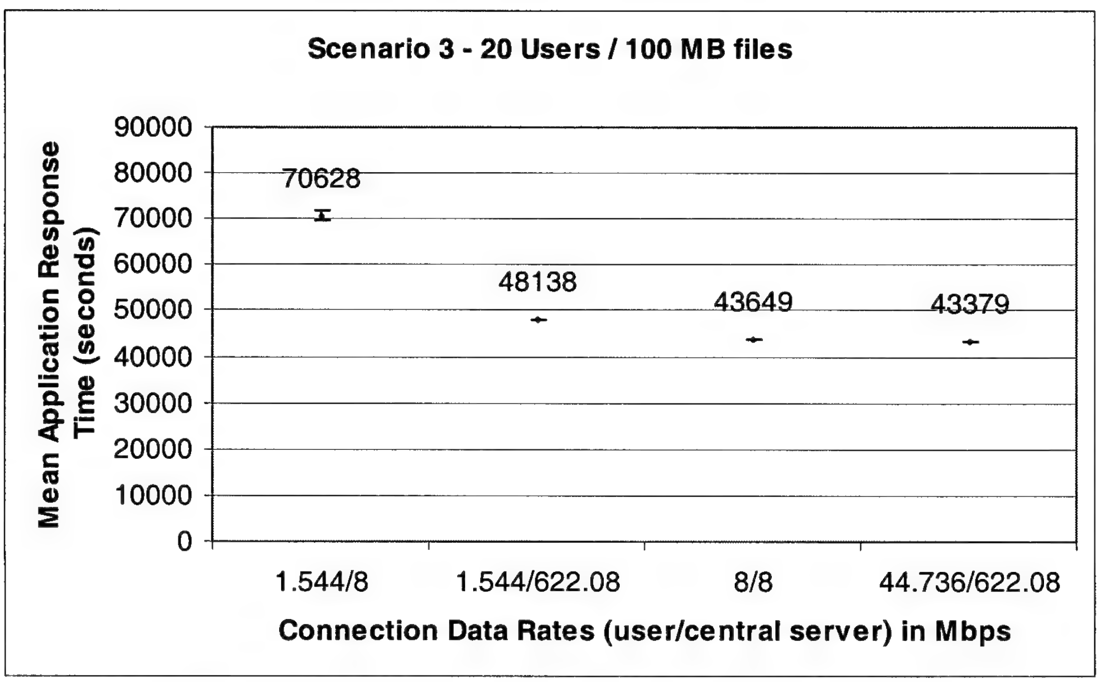


Figure A21. Scenario 3 - visual test for 20 users/100MB files

Appendix B: Data

Table B1. Scenario 2 data

	Scenario 2			
	T1		8Mbps/44.736Mbps	
	8Mbps	OC12	8Mbps	OC12
2 Users	6533.038	6261.227	6151.010	6151.002
2 Users	6534.441	6257.527	6151.010	6151.002
2 Users	6524.254	6264.327	6151.015	6151.002
2 Users	6511.330	6259.727	6151.015	6151.002
2 Users	6533.430	6262.727	6151.013	6151.002
10 Users	9614.476	6272.571	6151.172	6151.004
10 users	9613.565	6269.414	6151.214	6151.004
10 Users	9611.381	6272.228	6151.114	6151.004
10 Users	9612.544	6271.531	6151.117	6151.004
10 Users	9620.239	6270.468	6151.156	6151.004
20 Users	10796.347	6283.083	7209.992	6151.007
20 Users	10798.109	6283.010	7211.562	6151.007
20 Users	10792.793	6283.092	7210.452	6151.007
20 Users	10798.333	6283.579	7207.573	6151.007
20 Users	10794.057	6283.269	7206.732	6151.007

Table B2. Scenario 1 data

Scenario		Baseline Scenario			
User data rate		1.544Mbps		8Mbps/44.736Mbps	
C.S. data rate		1.544Mbps	622.08Mbps	8Mbps	622.08Mbps
1MB	2	502.386	481.156	434.050	433.407
1MB	2	501.377	479.355	434.252	433.407
1MB	2	501.178	480.773	434.259	433.407
1MB	2	501.180	481.066	434.355	433.407
1MB	2	502.080	482.156	434.150	433.407
1MB	10	753.318	533.354	547.735	509.949
1MB	10	754.804	533.676	547.206	509.949
1MB	10	754.343	534.200	547.560	509.949
1MB	10	755.321	533.499	546.244	509.949
1MB	10	754.203	533.537	546.815	509.949
1MB	20	997.217	671.885	802.590	662.192
1MB	20	998.436	672.058	802.655	662.192
1MB	20	997.682	672.148	802.930	662.192
1MB	20	997.372	671.955	802.895	662.192
1MB	20	998.381	672.101	802.982	662.192
10MB	2	5070.975	4859.656	4465.755	4337.607
10MB	2	5073.975	4862.556	4464.855	4337.607
10MB	2	5068.575	4860.856	4464.755	4337.607
10MB	2	5065.575	4859.483	4465.353	4337.607
10MB	2	5071.880	4855.056	4465.460	4337.607
10MB	10	7692.719	4840.057	5454.725	4337.610
10MB	10	7695.222	4838.537	5453.942	4337.610
10MB	10	7696.260	4837.553	5455.084	4337.610
10MB	10	7696.961	4839.422	5452.701	4337.609
10MB	10	7694.161	4838.661	5454.241	4337.609
10MB	20	9823.688	4823.799	7505.217	4337.812
10MB	20	9822.087	4823.206	7510.145	4337.812
10MB	20	9823.880	4823.688	7509.117	4337.812
10MB	20	9820.856	4823.986	7507.150	4337.812
10MB	20	9823.688	4822.945	7510.299	4337.812
100MB	2	50724.280	48643.706	44766.950	43378.607
100MB	2	50693.675	48630.441	44768.150	43378.607
100MB	2	50733.978	48639.061	44766.050	43378.607
100MB	2	50744.580	48626.878	44766.050	43378.607
100MB	2	50739.675	48639.456	44768.451	43378.607
100MB	10	78279.209	48585.456	54722.057	43378.609
100MB	10	78368.269	48584.899	54731.493	43378.609
100MB	10	78271.985	48587.681	54719.895	43378.609
100MB	10	78352.108	48589.157	54726.837	43378.610
100MB	10	78368.269	48588.621	54731.493	43378.610
100MB	20	78865.494	48379.914	56286.990	43378.810
100MB	20	78875.202	48374.717	56272.212	43378.810
100MB	20	78964.370	48371.294	56280.533	43378.810
100MB	20	78964.370	48375.276	56277.769	43378.810
100MB	20	78964.370	48375.239	56277.155	43378.810

Table B3. Scenario 3 data

Scenario		Scenario 3			
User data rate		1.544Mbps		8Mbps/44.736Mbps	
C.S. data rate		8Mbps	622.08Mbps	8Mbps	622.08Mbps
1MB	2	507.875	476.456	433.455	433.407
1MB	2	506.975	477.468	433.463	433.407
1MB	2	506.575	475.356	433.455	433.407
1MB	2	508.775	476.457	433.455	433.407
1MB	2	508.580	476.756	433.456	433.407
1MB	10	693.878	465.937	433.695	433.409
1MB	10	699.033	465.120	433.699	433.409
1MB	10	699.045	464.890	433.695	433.409
1MB	10	697.840	465.057	433.697	433.409
1MB	10	696.673	465.563	433.718	433.409
1MB	20	765.684	454.074	504.216	433.412
1MB	20	764.045	454.189	504.817	433.413
1MB	20	764.465	453.999	505.002	433.412
1MB	20	764.643	454.075	504.494	433.413
1MB	20	764.457	453.968	504.063	433.413
10MB	2	5097.478	4832.256	4337.655	4337.607
10MB	2	5089.950	4833.755	4337.655	4337.607
10MB	2	5095.575	4828.056	4337.655	4337.607
10MB	2	5091.079	4833.556	4337.655	4337.607
10MB	2	5103.875	4833.241	4337.655	4337.607
10MB	10	7041.543	4818.794	4338.011	4337.610
10MB	10	7043.178	4816.097	4337.958	4337.609
10MB	10	7034.288	4819.817	4337.940	4337.609
10MB	10	7036.634	4819.217	4337.964	4337.609
10MB	10	7042.602	4818.398	4337.960	4337.610
10MB	20	7819.599	4808.177	5038.031	4337.612
10MB	20	7815.523	4808.080	5041.597	4337.612
10MB	20	7821.387	4808.730	5040.936	4337.612
10MB	20	7814.278	4806.987	5040.467	4337.613
10MB	20	7813.350	4807.288	5037.722	4337.612
100MB	2	50971.680	48374.556	43378.655	43378.607
100MB	2	50993.380	48378.256	43378.656	43378.607
100MB	2	50970.480	48362.056	43378.659	43378.607
100MB	2	50929.486	48381.956	43378.655	43378.608
100MB	2	50975.986	48364.861	43378.660	43378.607
100MB	10	71057.079	48322.700	43379.602	43378.610
100MB	10	71115.674	48324.217	43379.642	43378.609
100MB	10	71103.135	48321.756	43379.623	43378.610
100MB	10	71029.277	48313.679	43379.646	43378.609
100MB	10	71087.509	48318.918	43379.530	43378.610
100MB	20	70654.964	48144.857	43646.620	43378.609
100MB	20	70672.406	48133.538	43651.569	43378.610
100MB	20	70556.366	48141.298	43650.166	43378.610
100MB	20	70628.604	48133.094	43646.580	43378.610
100MB	20	70581.652	48136.877	43649.942	43378.610

Appendix C: Performance Analysis Charts

Table C1. ANOVA for baseline scenario (all factors)

Baseline results only			f-val	significant
SSY=	1.81174E+11			
SS0=	70878337538			
SST=	1.10296E+11			
SSB=	1.01439E+11	0.9197038	4872.39	yes
SSC=	573594582.9	0.0052005	27.5512	yes
SSD=	778998345.6	0.0070628	74.8344	yes
SSF=	1423738042	0.0129084	136.771	yes
SSBC=	701504472.4	0.0063602	16.8475	yes
SSBD=	1147778867	0.0104064	55.1306	yes
SSBF=	1869532890	0.0169502	89.7982	yes
SSCD=	97998276.67	0.0008885	4.7071	yes
SSCF=	533139275.8	0.0048337	25.608	yes
SSDF=	231056975.1	0.0020949	22.1965	yes
Total % explained variation		0.9864094		
SSE=	1498986547	0.0135906	MSE=	10409629

In the chart above, factor B is the file size, factor C is the number of users, factor D is the user connection bandwidth, and factor F is the central server connection bandwidth. The remaining ANOVA charts in this appendix only consider the user connection bandwidth and central server connection bandwidth factors. Each chart represents a different configuration as identified by the chart's title. For each of the following analyses, factor A is the user connection bandwidth and factor B is the central server connection bandwidth.

Table C2. ANOVA for baseline – 2 users/1MB files

This ANOVA is for the baseline scenario consisting of 2 users downloading 1MB files							
Sum of Squares		% var.	DOF	Mean Sq. Val	Calc. F-vals	F-vals	Sig.?
SSY =	4296463.298						
SS0 =	2852581.622						
SST =	1443881.676						
SSE =	5.372442398		DOF=24				
MSE =	0.223851767						
SSA =	501844.2118	0.347566	DOF=2	250922.1059	1120929.76	2.53-2.59	yes
SSB =	386.8297025	0.0002679	DOF=1	386.8297025	1728.06187	2.92-2.97	yes
SSAB =	941645.2624	0.6521623	DOF=2	470822.6312	2103278.6	2.53-2.59	yes
% explained variation =		0.9999963					
% unexplained variation =		3.721E-06					

Table C3. ANOVA for baseline – 2 users/10MB files

This ANOVA is for the baseline scenario consisting of 2 users downloading 10MB files							
Sum of Squares		% var	DOF	Mean Sq. Val	calc F-vals	F-vals	Sig.?
SSY =	440375069.2						
SS0 =	292424003.5						
SST =	147951065.7						
SSE =	73.36391049		DOF=24				
MSE =	3.056829604						
SSA =	50957124.27	0.3444188	DOF=2	25478562.14	8334963.16	2.53-2.59	yes
SSB =	95374.21261	0.0006446	DOF=1	95374.21261	31200.3693	2.92-2.97	yes
SSAB =	96898493.88	0.6549361	DOF=2	48449246.94	15849508.5	2.53-2.59	yes
% explained variation =		0.9999995					
% unexplained variation =		4.959E-07					

Table C4. ANOVA for baseline – 2 users/100MB files

This ANOVA is for the baseline scenario consisting of 2 users downloading 100MB files							
Sum of Squares		% var	D.O.F.	Mean Sq. Val	Comp F-value	F-vals	Sig.?
SSY =	44122520396						
SS0 =	29299651066						
SST =	14822869330						
SSE =	1835.994422		24				
MSE =	76.4997676						
SSA =	5097434621	0.3438899	2	2548717310	33316667.4	2.53-2.59	yes
SSB =	10091143.94	0.0006808	1	10091143.94	131910.779	2.92-2.97	yes
SSAB =	9715341730	0.6554292	2	4857670865	63499158.5	2.53-2.59	yes
% explained variation =		0.9999999					
% unexplained variation =		1.239E-07					

Table C5. ANOVA for baseline – 10 users/1MB files

This ANOVA is for the baseline scenario consisting of 10 users downloading 1MB files							
Sum of Squares		% var	D.O.F.	Mean Sq. Vals	Comp f-vals	F-vals	Sig.?
SSY =	7066410.598						
SS0 =	4582958.577						
SST =	2483452.021						
SSE =	4.0898276		24				
MSE =	0.170409483						
SSA =	963178.6772	0.3878386	2	481589.3386	2826071.23	2.53-2.59	yes
SSB =	55430.27511	0.0223198	1	55430.27511	325276.939	2.92-2.97	yes
SSAB =	1464838.979	0.5898399	2	732419.4894	4297997.24	2.53-2.59	yes
% explained variation =		0.9999984					
% unexplained variation =		1.647E-06					

Table C6. ANOVA for baseline – 10 users/10MB files

This ANOVA is for the baseline scenario consisting of 10 users downloading 10MB files							
Sum of Squares		% var	D.O.F.	Mean Sq. Vals	Comp f-val	F-vals	Sig.?
SSY =	655954691.9						
SS0 =	415362534						
SST =	240592157.9						
SSE =	18.32175159		24				
MSE =	0.763406316						
SSA =	98790965.55	0.4106159	2	49395482.78	64704053	2.53-2.59	yes
SSB =	13152269.58	0.0546662	1	13152269.58	17228400.3	2.92-2.97	yes
SSAB =	128648904.4	0.5347178	2	64324452.21	84259785.2	2.53-2.59	yes
% explained variation =		0.9999999					
% unexplained variation =		7.615E-08					

Table C7. ANOVA for baseline – 10 users/100MB files

This ANOVA is for the baseline scenario consisting of 10 users downloading 100MB files							
Sum of Squares		% var	D.O.F.	Mean Sq. Vals	Comp f-vals	F-vals	Sig.?
SSY =	66863312695						
SS0 =	42195036037						
SST =	24668276659						
SSE =	9470.261643		24				
MSE =	394.5942351						
SSA =	10265284241	0.416133	2	5132642121	13007392.6	2.53-2.59	yes
SSB =	1406890839	0.057032	1	1406890839	3565411.54	2.92-2.97	yes
SSAB =	12996092108	0.526834	2	6498046054	16467666	2.53-2.59	yes
% explained variation =		0.9999996					
% unexplained variation =		3.839E-07					

Table C8. ANOVA for baseline – 20 users/1MB files

This ANOVA is for the baseline scenario consisting of 20 users downloading 1MB files							
Sum of Squares		% var	D.O.F.	Mean Sq. Vals	Comp f-vals	F-vals	Sig.?
SSY =	12651332.75						
SS0 =	8189400.634						
SST =	4461932.116						
SSE =	1.447535598		24				
MSE =	0.060313983						
SSA =	1489078.833	0.3337296	2	744539.4163	12344391.4	2.53-2.59	yes
SSB =	181279.2638	0.040628	1	181279.2638	3005592.63	2.92-2.97	yes
SSAB =	2791572.572	0.6256421	2	1395786.286	23142001.4	2.53-2.59	yes
% explained variation =		0.9999997					
% unexplained variation =		3.244E-07					

Table C9. ANOVA for baseline – 20 users/10MB files

This ANOVA is for the baseline scenario consisting of 20 users downloading 10MB files							
Sum of Squares		% vars	D.O.F.	Mean Sq. Vals	Comp f-vals	F-vals	Sig.?
SSY =	974735229.1						
SS0 =	584879876.6						
SST =	389855352.5						
SSE =	26.63366651		24				
MSE =	1.109736105						
SSA =	139391281.6	0.3575462	2	69695640.78	62803796.8	2.53-2.59	yes
SSB =	55622566.45	0.1426749	1	55622566.45	50122336.5	2.92-2.97	yes
SSAB =	194841477.9	0.4997789	2	97420738.95	87787302.4	2.53-2.59	yes
% explained variation =		0.9999999					
% unexplained variation =		6.832E-08					

Table C10. ANOVA for baseline – 20 users/100MB files

This ANOVA is for the baseline scenario consisting of 20 users downloading 100MB files							
Sum of Squares		% var	D.O.F.	Mean Sq. Vals	Comp f-vals	F-vals	Sig.?
SSY =	68093218031						
SS0 =	42925622276						
SST =	25167595755						
SSE =	10810.13017		24				
MSE =	450.4220904						
SSA =	10211505355	0.4057402	2	5105752677	11335484.6	2.53-2.59	yes
SSB =	1573367590	0.0625156	1	1573367590	3493095.97	2.92-2.97	yes
SSAB =	13382712001	0.5317438	2	6691356000	14855745.6	2.53-2.59	yes
% explained variation =		0.9999996					
% unexplained variation =		4.295E-07					

The next set of ANOVA charts are for scenario 2. The first chart is the ANOVA for all factors where factor A is the number of users, factor B is the user connection and factor C is the central server connection. The remainder of the ANOVAs for scenario 2 only consider the user connection and central server connection bandwidth factors (factors B and C).

Table C11. ANOVA for scenario 2

This ANOVA is for scenario 2							
Sum of Squares		% var	D.O.F.	Mean Sq. Vals	Comp f-val	F-vals	Sig. (90%)?
SSY =	3053767860						
SS0 =	2920269820						
SST =	133498040.9						
SSE =	8185794.135		D.O.F.=48	170537.3778			
MSE =	170537.3778						
SSA =	18031322.86	0.1350681	D.O.F.=2	9015661.429	52.8661901	2.44	yes
SSB =	25277128.94	0.1893446	D.O.F.=1	25277128.94	148.220462	2.84	yes
SSC =	35117537.79	0.2630566	D.O.F. = 1	35117537.79	205.92282	2.84	yes
SSAB =	8358101.772	0.0626084	D.O.F. = 2	4179050.886	24.5051902	2.44	yes
SSAC =	17738244.13	0.1328727	D.O.F. = 2	8869122.066	52.00691	2.44	yes
SSBC =	20789911.27	0.155732	D.O.F.=1	20789911.27	121.908238	2.84	yes
% explained variation =		0.9386823					
% unexplained variation =		0.0613177					

Table C12. ANOVA for scenario 2 – 2 users

This ANOVA is for scenario 2 - user and central server connection speeds only for 2 users							
Sum of Squares		% var	D.O.F.	Mean Sq. Vals	Comp f-vals	F-vals	Sig. (90%)?
SSY =	787384761						
SS0 =	524607654.8						
SST =	262777106.1						
SSE =	413.5424565		24	17.23093569			
MSE =	8.615467844						
SSB =	73425093.65	0.2794197	2	36712546.83	2130618.3	2.53	yes
SSC =	59053.0094	0.0002247	1	59053.0094	3427.15047	2.92	yes
SSBC =	189292545.9	0.720354	2	94646272.97	5492810.99	2.53	yes
% explained variation =		0.9999984					
% unexplained variation =		1.574E-06					

Table C13. ANOVA for scenario 2 – 10 users

This ANOVA is for scenario 2 - user and central server connection speeds only for 10 users							
Sum of Squares		% var	D.O.F.	Mean Sq. Vals	Comp f-vals	F-vals	Sig. (90%)?
SSY =	1037187605						
SS0 =	662128697.2						
SST =	375058907.5						
SSE =	54.11712636		24	2.254880265			
MSE =	1.127440132						
SSB =	157937528.3	0.4211006	2	78968764.17	4582964.36	2.53	yes
SSC =	9314986.561	0.0248361	1	9314986.561	540596.676	2.92	yes
SSBC =	207806338.4	0.5540632	2	103903169.2	6030036.39	2.53	yes
% explained variation =		0.9999999					
% unexplained variation =		1.443E-07					

Table C14. ANOVA for scenario 2 – 20 users

This ANOVA is for scenario 2 - user and central server connection speeds only for 20 users							
Sum of Squares		% var	D.O.F.	Mean Sq. Vals	Comp f-vals	f-vals	Sig. (90%)?
SSY =	1229195495						
SS0 =	772131076.3						
SST =	457064418.6						
SSE =	40.74747065		24	1.697811277			
MSE =	0.848905638						
SSB =	181631874.8	0.3973879	2	90815937.41	53490007.2	2.53	yes
SSC =	25863148.37	0.0565853	1	25863148.37	15233229.2	2.92	yes
SSBC =	249569354.7	0.5460267	2	124784677.3	73497378.1	2.53	yes
% explained variation =		0.9999999					
% unexplained variation =		8.915E-08					

Table C15. ANOVA for scenario 3

B = file size (3)			
C = # Users (3)			
D = User connection speed (2)			
F = C.S. connection speed (2)			
Scenario 3		comp. f-value	sig (90%)?
SSY=	1.56359E+11		
SS0=	61459929147		
SST=	94898689461		
SSB=	89133902298	0.9392532	4871.342085	yes
SSC=	144455193.8	0.0015222	7.894758861	yes
SSD=	1011255851	0.0106562	110.5342201	yes
SSF=	419342260.3	0.0044188	45.83574932	yes
SSBC=	192064609.7	0.0020239	5.248353276	yes
SSBD=	1480936920	0.0156055	80.93609903	yes
SSBF=	561089690.6	0.0059125	30.66464895	yes
SSCD=	131376753.1	0.0013844	7.179996498	yes
SSCF=	118801419	0.0012519	6.492729897	yes
SSDF=	388036771.9	0.004089	42.41393698	yes
Total % explained variation		0.9861175		
SSE=	1317427693	0.0138825	MSE=	9148803.42

Table C16. ANOVA for scenario 3 – 2 users/1MB files

This ANOVA is for scenario 3 consisting of 2 users downloading 1MB files							
Sum of Squares		% var	D.O.F.	Mean Sq. Vals	Comp f-vals	f-vals	Sig (90%)?
SSY =	4302972.345						
SS0 =	2855532.166						
SST =	1447440.18						
SSE =	6.051579835		D.O.F.=24				
MSE =	0.25214916						
SSA =	505674.7135	0.3493579	D.O.F.=2	252837.3567	1002729.325	2.59	yes
SSB =	816.7894602	0.0005643	D.O.F.=1	816.7894602	3239.310656	2.97	yes
SSAB =	940942.6251	0.6500736	D.O.F.=2	470471.3126	1865845.252	2.59	yes
% explained variation =		0.9999958					
% unexplained variation =		4.181E-06					

Table C17. ANOVA for scenario 3 – 10 users/1MB files

This ANOVA is for scenario 3 consisting of 10 users downloading 1MB files							
Sum of Squares		% var	D.O.F.	Mean Sq. Vals	Comp f-vals	f-vals	Sig (90%)?
SSY =	5393394.877						
SS0 =	3433126.561						
SST =	1960268.316						
SSE =	19.17774353		D.O.F.=24				
MSE =	0.799072647						
SSA =	885861.3285	0.4519082	D.O.F.=2	442930.6643	1756621.615	2.59	yes
SSB =	44958.56833	0.0229349	D.O.F.=1	44958.56833	178301.4798	2.97	yes
SSAB =	1029429.241	0.5251471	D.O.F.=2	514714.6207	2041310.077	2.59	yes
% explained variation =		0.9999902					
% unexplained variation =		9.783E-06					

Table C18. ANOVA for scenario 3 – 20 users/1MB files

This ANOVA is for scenario 3 consisting of 20 users downloading 1MB files							
Sum of Squares		% var	D.O.F.	Mean Sq. Vals	Comp f-vals	f-vals	Sig (90%)?
SSY =	6166300.755						
SS0 =	3875951.938						
SST =	2290348.817						
SSE =	2.158265921		D.O.F.=24				
MSE =	0.089927747						
SSA =	943205.8298	0.4118175	D.O.F.=2	471602.9149	1870333.081	2.59	yes
SSB =	121414.7952	0.0530115	D.O.F.=1	121414.7952	481519.7294	2.97	yes
SSAB =	1225726.034	0.53517	D.O.F.=2	612863.0169	2430557.442	2.59	yes
% explained variation =		0.9999991					
% unexplained variation =		9.423E-07					

Table C19. ANOVA for scenario 3 – 2 users/10MB files

This ANOVA is for scenario 3 consisting of 2 users downloading 10MB files							
Sum of Squares		% var	D.O.F.	Mean Sq. Vals	Comp f-vals	F-vals	Sig (90%)?
SSY =	434725306.6						
SS0 =	288393823.8						
SST =	146331482.8						
SSE =	146.8744003		D.O.F.=24				
MSE =	6.11976668						
SSA =	52082641.77	0.3559223	D.O.F.=2	26041320.88	103277444.6	2.59	yes
SSB =	57845.68251	0.0003953	D.O.F.=1	57845.68251	229410.5702	2.97	yes
SSAB =	94190848.47	0.6436814	D.O.F.=2	47095424.24	186776050.6	2.59	yes
% explained variation =		0.999999					
% unexplained variation =		1.004E-06					

Table C20. ANOVA for scenario 3 – 10 users/10MB files

This ANOVA is for scenario 3 consisting of 10 users downloading 10MB files							
Sum of Squares		% var	D.O.F.	Mean Sq. Vals	Comp f-vals	f-vals	Sig (90%)?
SSY =	552035416.6						
SS0 =	351360342.1						
SST =	200675074.5						
SSE =	70.71132601		D.O.F.=24				
MSE =	2.94630525						
SSA =	94258823.18	0.4697087	D.O.F.=2	47129411.59	186910841.3	2.59	yes
SSB =	4112706.652	0.0204944	D.O.F.=1	4112706.652	16310610.18	2.97	yes
SSAB =	102303473.9	0.5097966	D.O.F.=2	51151736.96	202863007.7	2.59	yes
% explained variation =		0.9999996					
% unexplained variation =		3.524E-07					

Table C21. ANOVA for scenario 3 – 20 users/10MB files

This ANOVA is for scenario 3 consisting of 20 users downloading 10MB files							
Sum of Squares		% var	D.O.F.	Mean Sq. Vals	Comp f-vals	f-vals	Sig (90%)?
SSY =	642161078.3						
SS0 =	403408239.3						
SST =	238752839						
SSE =	63.15644311		D.O.F.=24				
MSE =	2.631518463						
SSA =	105583032.7	0.4422273	D.O.F.=2	52791516.35	209366219.5	2.59	yes
SSB =	11476966.47	0.0480705	D.O.F.=1	11476966.47	45516576.29	2.97	yes
SSAB =	121692776.7	0.5097019	D.O.F.=2	60846388.33	241311089	2.59	yes
% explained variation =		0.9999997					
% unexplained variation =		2.645E-07					

Table C22. ANOVA for scenario 3 – 2 users/100MB files

This ANOVA is for scenario 3 consisting of 2 users downloading 100MB files							
Sum of Squares		% var	D.O.F.	Mean Sq. Vals	Comp f-vals	f-vals	Sig (90%)?
SSY =	43505262777						
SS0 =	28860326977						
SST =	14644935799						
SSE =	2504.828032		D.O.F.=24				
MSE =	104.3678347						
SSA =	5219558395	0.3564071	D.O.F.=2	2609779197	10350140367	2.59	yes
SSB =	5615647.36	0.0003835	D.O.F.=1	5615647.36	22271132.55	2.97	yes
SSAB =	9419759252	0.6432093	D.O.F.=2	4709879626	18678942377	2.59	yes
% explained variation =		0.9999998					
% unexplained variation =		1.71E-07					

Table C23. ANOVA for scenario 3 – 10 users/100MB files

This ANOVA is for scenario 3 consisting of 10 users downloading 100MB files							
Sum of Squares		% var	D.O.F.	Mean Sq. Vals	Comp f-vals	f-vals	Sig (90%)?
SSY =	55752501178						
SS0 =	35417259475						
SST =	20335241703						
SSE =	5020.667387		D.O.F.=24				
MSE =	209.1944745						
SSA =	9631652734	0.4736434	D.O.F.=2	4815826367	19099117249	2.59	yes
SSB =	431654035.4	0.0212269	D.O.F.=1	431654035.4	1711899559	2.97	yes
SSAB =	10271929913	0.5051295	D.O.F.=2	5135964956	20368756971	2.59	yes
% explained variation =		0.9999998					
% unexplained variation =		2.469E-07					

Table C24. ANOVA for scenario 3 – 20 users/100MB files

This ANOVA is for scenario 3 consisting of 20 users downloading 100MB files							
Sum of Squares		% var	D.O.F.	Mean Sq. Vals	Comp f-vals	f-vals	Sig (90%)?
SSY =	55456070184						
SS0 =	35289320697						
SST =	20166749486						
SSE =	9679.037147		D.O.F.=24				
MSE =	403.2932145						
SSA =	9435924416	0.4678952	D.O.F.=2	4717962208	18710997141	2.59	yes
SSB =	431348787.9	0.0213891	D.O.F.=1	431348787.9	1710688976	2.97	yes
SSAB =	10299466603	0.5107153	D.O.F.=2	5149733302	20423360942	2.59	yes
% explained variation =		0.9999995					
% unexplained variation =		4.8E-07					

Table C25. ANOVA for scenarios 1 & 3

Comparison of scenarios 1 and 3 only							
Sum of Squares		% var	D.O.F.	Mean Sq. Vals	Comp f-val	f-vals	Sig?
SSY =	8599435.643						
SS0 =	5708113.406						
SST =	2891322.237						
SSE =	110.7266578		24				
MSE =	1.537870247						
SSA =	0.381286593	1.31873E-07	1	0.381286593	0.247931575	2.97	no
SSB =	1007514.925	0.348461653	2	503757.4627	327568.2482	2.59	yes
SSC =	1163.911373	0.000402553	1	1163.911373	756.8332734	2.97	yes
SSAB =	3.99984636	1.3834E-06	2	1.99992318	1.300449881	2.59	no
SSAC =	39.70778995	1.37334E-05	2	19.85389497	12.90999356	2.59	yes
SSBC =	1882488.585	0.651082249	2	941244.2924	612044.0229	2.59	yes
% explained variation =		0.999961704					
% unexplained variation =		3.82962E-05					

Table C26. ANOVA for all scenarios

Comparison of all three scenarios							
Sum of Squares		% var	D.O.F.	Mean Sq. Vals	Comp f-vals	f-vals	Sig?
SSY =	795984196.6						
SS0 =	230267219.2						
SST =	565716977.5						
SSE =	109026422.5		24				
MSE =	1514255.867						
SSA =	300048549.5	0.530386326	2	150024274.7	99.07458703	2.59	yes
SSB =	33255788.85	0.058785206	2	16627894.42	10.98090143	2.59	yes
SSC =	28276.64023	4.99837E-05	1	28276.64023	0.018673621	2.97	no
SSAB =	41176823.73	0.072786968	4	10294205.93	6.798194515	2.25	yes
SSAC =	31979.98833	5.653E-05	2	15989.99417	0.010559638	2.59	no
SSBC =	82149136.33	0.145212429	2	41074568.17	27.12524947	2.59	yes
% explained variation =		0.807277443					
% unexplained variation =		0.192722557					

Appendix D: Task Configuration Tables

Table D1. Task configuration table for scenario 2

(Manual Configuration) Table

Phase Name	Start Phase After	Source	Destination	REQ/RESP Patt	End Phase When
task 1	Application Starts	User_1	Central Server	REQ->REQ->...	Final Request Arrives
task 2	Previous Phase Ends	Central Server	User_1	REQ->REQ->...	Final Request Arrives

2 Rows Delete Insert Duplicate Move Up Move Down

Details Promote Cancel OK

Table D2. Task configuration table for scenario 3

(Manual Configuration) Table

Phase	Start Phase After	Source	Destination	End Phase When
task 1	Application Starts	User_1	Central Server	Final Request Arrives at Destination
task 2	Previous Phase Ends	Central Server	User_1	Final Request Arrives at Destination
task 3	Application Starts	User_1	RS_1	Final Response Arrives at Source
task 4	Application Starts	User_1	RS_2	Final Response Arrives at Source
task 5	Application Starts	User_1	RS_3	Final Response Arrives at Source

5 Rows Delete Insert Duplicate Move Up Move Down

Details Promote Cancel OK

Bibliography

[ATM00] "Why all of the interest in ATM?," Available at http://www.atmforum.com/atmforum/library/notes2.html, August 2000.

[BAE00] Baenen, Eric. Virtual Distributed Laboratory Project Manager, Veridian, Engineering Division, Wright-Patterson AFB OH. personal interview. 15 November 2000.

[BEG00] Begole, James, Craig Struble, and Clifford Shaffer. "Leveraging Java Applets: Toward Collaboration Transparency in Java," IEEE Internet Computing, 1: 57-64 (March-April 1997)

[DRE00] "Defense Research and Engineering Network (DREN Executive Summary," Available at http://www.arl.mil/HPCMP/DREN/drenexe3.html, June 2000.

[DYK00] Dykstra, Phil and Cindy Dykstra. "Research for the Defense Research and Engineering Network (DREN)," Network Analysis Times, I: 5, Available at http://moat.nlanr.net/NATimes.htm, (January 2000).

[FAR98] Farley, Jim. Java Distributed Computing. Sebastopol: O'Reilly & Associates, Inc., 1998.

[FOS95] Foster, Ian T. Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering. Reading: Addison-Wesley Publishing Company, 1995.

[GAD97] Gadecki, Cathy and Christine Heckart. ATM For Dummies. Foster City: IDG Books Worldwide, 1997.

[GEO00] Johnston, William and Sonia Sachs. "Distributed, Collaboratory Experiment Environments (DCEE) Program: Overview and Final Report, February, 1997," Available at http://george.lbl.gov/DCEEpage/DCEE/Overview.html, August 2000.

[HAR00] Harmer, Paul. VDL Program Manager, Wright-Patterson AFB OH. "File sizes." Electronic Message. 28 November 2000.

[HPC00] "High Performance Computing," Available at http://www.asc.hpc.mil/hpcomputing/index.htm, November 2000.

[IUK00] "What is Asynchronous Transfer Mode (ATM)?," Available at the Indiana University Knowledge Base site located at http://kb.indiana.edu/data/aequ.html, August 00.

[JAI91] Jain, Raj. The Art of Computer Systems Performance Analysis. New York: John Wiley & Sons, Inc., 1991.

[KUM94] Kumar, Vipin and others. Introduction to Parallel Computing: Designand Analysis of Algorithms. Redwood City: The Benjamin/Cummings Publishing Company, Inc., 1994.

[MAH00] Mahmoud, Qusay H. Distributed Programming with Java. Greenwich: Manning Publications Co., 2000.

[NAP00] "Direct Connect OC3 Service," Available at http://www.nap.net/services/s_oc3.shtml, August 2000.

[OPN00] OPNET Technologies, Inc. OPNET Modeler Training Manual for AFRL. Part Number D00044, Version 1, Release: Modeler 7.0.B, PL4_1083, Jun00 Models. Washington DC: OPNET Technologies, Inc., 2000.

[REL00] Model library update release notes, available at http://www.opnet.com, December 2000.

[SUP00] Sunderam, V. and others, "CCF: Collaborative Computing Frameworks," Department of Mathematics and Computer Science, Emory University, Atlanta GA, available at http://www.supercomp.org/sc98/techpapers/sc98_fullabstracts/sunderam1290/index.htm, July 2000.

[TAN95] Tanenbaum, Andrew S. Distributed Operating Systems. New Jersey: Prentice Hall, 1995.

[VDL00] "The VDL Central Library Capability Description," Available at http://www.vdl.afrl.af.mil/mission/index.htm, 2000.

[WAR00] Ward, Jason T. Enhancing a Virtual Distributed Library User Interface via Server-Side User Profile Caching. MS thesis, AFIT/GCS/ENG/00M-23, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 2000.

Vita

First Lieutenant David B. Hooten was born in 1967 in Torrance, California. He graduated from Mulvane High School in Mulvane, Kansas in May 1985. In February 1988, he enlisted in the United States Air Force and served as an airborne command post electronics communications technician with the 55th Maintenance Squadron, Offutt AFB, Nebraska. In November 1994, he was accepted into AFROTC at the University of Nebraska at Omaha, where he completed a Bachelor of General Studies in Computer Science and was commissioned in May 1997.

His first assignment as a Second Lieutenant was to the Air Force Weather Agency, Offutt AFB, Nebraska as a systems duty officer. In August 1999, he entered the Graduate School of Engineering and Management, Air Force Institute of Technology. Upon graduation, he will be assigned to the Air Force Logistics Management Agency.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED
20-03-01	MASTER'S THESIS	APR 00 - MAR 01

4. TITLE AND SUBTITLE

A TRAFFIC PATTERN-BASED COMPARISON OF BULK IMAGE REQUEST RESPONSE TIMES FOR A VIRTUAL DISTRIBUTED LABORATORY

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

DAVID B. HOOTEN, 1LT, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 P. Street, Bldg 640
Wright-Patterson AFB, OH 45433
DSN: 787-4215

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GCS/ENG/01M-03

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Virtual Distributed Laboratory
AFRL/SNAS
Attn: Captain Paul Harmer
2010 Fifth Street
WPAFB, OH 45433 DSN: 785-6329x2730 e-mail: paul.harmer@vdl.afrl.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AILABILITY STATEMENT

Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES

Advisor: Major Michael L. Talbert, AFIT/ENG, DSN: 785-6565x4280

14. ABSTRACT

Various agencies throughout the Department of Defense possess intelligence imagery and electrooptical signature data required by researchers in the field of automatic target recognition (ATR). The Air Force Research Laboratory, Sensors Directorate, has been tasked with creating a virtual distributed laboratory (VDL) which will make this data available to ATR researchers via high speed networks such as the defense research and engineering network (DREN). For this research, a model for simulating potential operational network configurations and collaboration scenarios was developed and implemented using OPNET. The results of the simulations were analyzed using statistical methods to determine the impact on performance of network configuration, connection speed, server capability, and data size. Connection speed proved to be the ultimate limiting factor on system performance, but statistical insights regarding usage patterns and file sizes are drawn from the results as well. This research provides VDL designers with performance trend data and enhances the design process by providing insight into how design decisions will affect future network performance.

15. SUBJECT TERMS

Distributed data processing, computer networks, ATM, Virtual Distributed Laboratory, automatic target recognition, defense research and engineering network

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Major Michael L. Talbert, ENG
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UL	138	(937)255-6565x4280